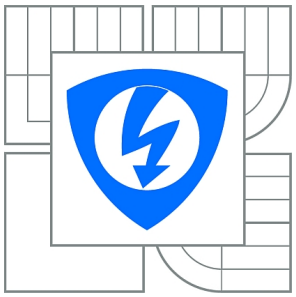

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY


FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

DEPARTMENT OF TELECOMMUNICATIONS


# PROUDOVÝ POSTRANNÍ KANÁL MIKROPROCESORŮ

SIDE CURRENT CHANNEL OF MICROPROCESSORS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS


AUTOR PRÁCE — Bc. ONDŘEJ OBRUČNÍK

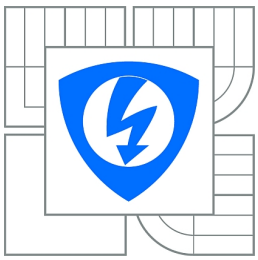
AUTHOR

VEDOUCÍ PRÁCE — Ing. ZDENĚK MARTINÁSEK

SUPERVISOR



BRNO 2010



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

**Fakulta elektrotechniky a komunikačních technologií**

**Ústav telekomunikací**

# Diplomová práce

magisterský navazující studijní obor
**Telekomunikační a informační technika**

| | | | |
|---|---|---|---|
| ***Student:*** | Bc. Ondřej Obručník | ***ID:*** | 78619 |
| ***Ročník:*** | 2 | ***Akademický rok:*** | 2009/2010 |

**NÁZEV TÉMATU:**

## Proudový postranní kanál mikroprocesorů

**POKYNY PRO VYPRACOVÁNÍ:**

Prostudujte základní útoky postranními kanály na kryptografický modul. Zaměřte se především na výkonový postranní kanál. Realizujte experimentální měření výkonového postranního kanálu u procesorů PIC. Procesor bude zpracovávat cyklicky jednu instrukci (př. XOR, ADDL, RRF, RLF atd.). Získané informace postraním kanálem přehledně zpracujte a porovnejte minimálně pro čtiři zvoloené operace.

**DOPORUČENÁ LITERATURA:**

[1] ALFRED J. MENEYES, Paul C. van Oorschot, Scott A. Vanstone, Handbook of Applied Cryptography, CRC Press, 1996
[2] KOCHER, P., JAFFE, J., JUN, B.: Introduction to Differential Power Analysis and Related Attacks, San Francisco, 1998. [.pdf dokument]. Dostupný z WWW: <http://www.cryptography.com/resources/ whitepapers/DPATechInfo.pdf>

| | | | |
|---|---|---|---|
| ***Termín zadání:*** | 29.1.2010 | ***Termín odevzdání:*** | 26.5.2010 |

***Vedoucí práce:*** Ing. Zdeněk Martinásek

**prof. Ing. Kamil Vrba, CSc.**
*Předseda oborové rady*

**UPOZORNĚNÍ:**

# ANOTACE

V této diplomové práci je podrobně prostudována problematika proudového (výkonového) postranního kanálu. Zabývá se útokem na tento typ postranního kanálu a způsoby, kterými je možné tento kanál analyzovat. Také jsou zde prezentovány dvě měřící metody, pomocí kterých je možné útok úspěšně provézt.

Dále práce popisuje postup, který byl použitý pro analýzu proudového postranního kanálu mikroprocesoru PIC16F84A. Zkoumaný mikroprocesor, zapojený do obvodu podle zde uvedeného schématu, postupně zpracovával různé programy realizující vždy jinou operaci, která odpovídá konkrétní použité instrukci. V odpovídajících kapitolách jsou také uvedeny výsledné hodnoty a grafy, které se podařilo měřením získat.

## Klíčová slova:

Proudový postranní kanál, výkonová analýza, mikroprocesor PIC16F84A, odporový bočník, měřící metoda, praktická realizace.

# ABSTRACT

In this masters's thesis is closely studied questins of current (power) side channel. It deals with attack upon this type of side channels and methods, which can this channel analyse. Also two methods of measurements, which make possible successfully attack, are presented here.

Below the work describes progress, which was used for analyse current side channel of chip PIC16F84A. This chip, which was plugged in the circuit in agreement with diagram introduced here, processes step by step variety of programs implementing always other operation, which matches concrete used instruction. In corresponding chapters are introduced resulting values and graphs, which was obtained by measurement.

## Keywords:

Current side channel, power analyse, chip PIC16F84A, shunt resistor, method of measurement, practical realization.

OBRUČNÍK, O. *Proudový postranní kanál mikroprocesorů* . Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2010. 64 s. Vedoucí diplomové práce Ing. Zdeněk Martinásek.

# Prohlášení

Prohlašuji, že svou diplomovou práci na téma Proudový postranní kanál mikroprocesorů jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne 26. 5. 2010 .....

podpis autora

# Poděkování

Děkuji vedoucímu práce ing. Zdeňkovi Martináskovi za velmi užitečnou metodickou pomoc a cenné rady při zpracování diplomové práce.


V Brně dne 26. 5. 2010 .....

podpis autora

# OBSAH

# ÚVOD

Diplomová práce se zabývá postranními kanály kryptografického modulu.

První kapitola shrnuje základní pojmy a definice z oblasti kryptologie. Nejprve je definován pojem kryptografický modul, následně jsou popsány vlastnosti kryptografických algoritmů a protokolů, které jsou nedílnou součástí každého kryptografického modulu. Dále je vysvětlen vznik postranních kanálů a hlavní rozdíl mezi klasickou kryptoanalýzou a kryptoanalýzou využívající postranní kanály. Práce interpretuje základní principy vzniku výkonového, elektromagnetického, časového a chybového postranního kanálu a zároveň popisuje typy analýz postranních kanálů, které se používají k útokům na kryptografické moduly.

Druhá kapitola se podrobně zabývá útokem proudovým (výkonovým) postranním kanálem, který využívá měření proudové spotřeby procesorů při vykonávání různých instrukcí, přičemž se uvnitř elektronického zařízení překlápí různý počet tranzistorů a tím dochází k nežádoucímu zanesení informace o vnitřním stavu programu do výkonové spotřeby.

Ve třetí a čtvrté kapitole jsou pak shrnuty vlastnosti procesorů s komplexní a redukovanou instrukční sadou, kde zvýšená pozornost je věnována především mikroprocesoru PIC16F84A, který je v této práci předmětem výzkumu.

V dalších kapitolách jsou pak popsány postupy nutné pro úspěšné provedení experimentálního měření proudového postranního kanálu u zkoumaného mikroprocesoru PIC16F84A a následné vyhodnocení.

# 1 ZÁKLADY KRYPTOANALÝZY

Kryptologie je věda zabývající se šifrováním a dešifrováním informací. Nesnaží se utajit samotnou existenci konkrétní informace, ale snaží se utajit její význam. Mezi hlavní odvětví kryptologie patří kryptografie a kryptoanalýza.

Kryptografie se zabývá šifrováním. Zkoumá způsoby a metody, jakými lze význam informace změnit do zdánlivě nesrozumitelné podoby. Zpráva je pak pomocí různých utajovacích metod zašifrována a bez znalosti tajného klíče ji nelze přečíst.

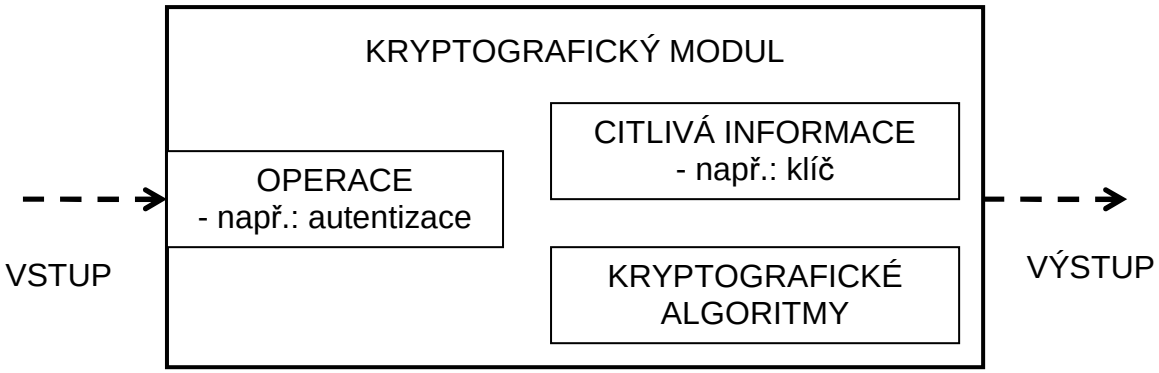
Kryptoanalýza je opak kryptografie. Je to označení pro vědu, která se zabývá rozluštěním šifrovaných zpráv či jiných informací bez znalosti tajného klíče, čímž také testuje odolnost kryptografického systému [2, 15].

## 1.1 KRYPTOGRAFICKÝ MODUL

Kryptografický modul (Obr. 1.1) je fyzická implementace konkrétního kryptografického algoritmu a používá se pro zajištění všech bezpečnostních požadavků. Jedná se o zařízení, které lze realizovat softwarově i hardwarově. Všechny citlivé operace a procesy, které souvisí například s autentizací, ověřováním, podepisováním, šifrováním nebo dešifrováním, probíhají uvnitř tohoto modulu. Požaduje se po něm rychlé plnění kryptografických služeb, přičemž s okolím může komunikovat jen po definovaných rozhraních. Činnost kryptografického modulu je označení pro všechny operace, které v něm probíhají. Bezpečnostní požadavky kryptografického modulu jsou:

- fyzická bezpečnost,
- logická bezpečnost,
- bezpečnost prostředí.

V praxi se kryptografické moduly objevují například ve formě počítačů, bankomatů, serverů, automatů nebo čipových karet [2, 9].




Obr. 1.1: Obecný model kryptografického modulu.

## 1.2 KRYPTOGRAFICKÉ ALGORITMY A PROKOLY

- **Kryptografické algoritmy.**

Patří mezi základní prvky kryptografie. Jsou důsledkem náročné teoretické přípravy, která řeší určitou kryptografickou problematiku. Jednotlivé algoritmy mají různé úkoly, proto jsou vytvářeny pro konkrétní situace, například šifrování u internetového bankovnictví. Kombinace jednotlivých algoritmů a použití kryptografických protokolů pak zajistí potřebné podmínky a prostředí pro činnost daného algoritmu, čímž dochází ke vzniku tzv. kryptosystému.

- **Kryptografické protokoly.**

Kryptografické zařízení obsahující kryptografické algoritmy spolu komunikují pomocí těchto protokolů. Ke komunikaci dojde v případě, že algoritmus potřebuje v jistém kroku odezvu druhé strany, ta pak probíhá v časové návaznosti. Popisují tedy kryptografické techniky a komunikaci, která probíhá mezi těmito technikami při spojení dvou či více zařízení.

V kryptografii mají kryptografické protokoly různorodé uplatnění. Pro testování spojení se používají ty jednodušší protokoly, naopak v oblasti bezpečné komunikace a elektronického bankovnictví se používají ty složitější.

Vlivem chybně navržených kryptografických protokolů dochází k prolomení kryptografických systémů. Chyba může nastat například v samotném protokolu, v jeho

implementaci nebo může být způsobena lidským faktorem. Zřídka kdy dochází k prolomení vinou kryptografického algoritmu [2].

### 1.2.1 ŠIFROVACÍ ALGORITMUS DES

Šifrovací algoritmus DES (Data Encryption Standard), označovaný také jako DEA (Data Encryption Algorithm), je symetrický blokový šifrovací algoritmus. Dnes již tento algoritmus není bezpečný a jeho použití se nedoporučuje. Jeho krátký 53 bitový klíč je možné rozluštit do 24 hodin. Ale i tak ho mnohé systémy stalé využívají.

DES šifruje data po 64 bitových blocích. A jelikož je to symetrický algoritmus, tak do algoritmu vstupuje 64 bitový blok otevřeného textu a poté z něho vystupuje 64 bitový blok šifrovaného textu. Pro šifrování i dešifrováni se tedy používá stejný klíč. Délka využitého klíče je 56 bitů, ten se sice většinou reprezentuje jako 64 bitový, ale každý osmý bit slouží pro paritní zabezpečení a při šifrování se ignoruje. DES je iterační šifrou a jednotlivé iterace se nazývají rundy [2, 15].

### 1.2.2 ŠIFROVACÍ ALGORITMUS AES

Šifrovací algoritmus AES (Advanced Encryption Standard) značí pokročilý šifrovací standard a je to nástupce algoritmu DES. Jeho původní název byl Rijndael. Je to symetrický šifrovací algoritmus s délkou klíče 128 až 256 bitů v krokování po 32 bitech, ten se pak aplikuje na bloky stejné délky. Velikost klíče a bloku je nezávislá. Šifrovací klíč se pak rozšiřuje na expandovaný klíč, jehož části se následně užívají pro šifrování bloku v jednotlivých rundách. Počet rund se pohybuje v rozmezí od 10 do 14 a je závislý přímo na délce bloku a délce klíče. Rozšíření potom závisí na délce klíče [14].

## 1.3 POSTRANNÍ KANÁL

Kryptografický modul při svém provozu vyzařuje různá záření a čerpá výkon zdroje, tyto děje můžou být spojeny s operacemi, které uvnitř kryptografického modulu probíhají. Postranní kanál je každá nežádoucí výměna informací mezi kryptografickým

modulem a okolím. Díky tomu lze zjistit, jakou činnost dané zařízení právě provádí a v jakém se nachází vnitřním stavu. V dnešní době neexistuje žádný konkrétní postup pro konstrukci kryptografického modulu, který by byl vůči postranním kanálům odolný. Proto také různé společnosti zkoumají možná zneužití různých postranních kanálů. Dnes už nedostačuje zvolit kvalitní šifru, ale je důležité věnovat velkou pozornost také její implementaci. Postranních kanálů existuje celá řada, mezi nejznámější patří:

- časový,
- proudový (výkonový, napěťový),
- elektromagnetický,
- chybový.

Princip časového postranního kanálu je založen na sledování času, který je nutný pro vykonání určitých kryptografických operací. Ty bývají závislé na zpracovávané informaci, tedy na otevřeném nebo šifrovaném textu či klíči.

Vznik proudového postranního kanálu je dán vlastností každého elektronického zařízení, což je potřeba spotřebovávat elektrický výkon ze zdroje. Je založen na sledování proudové spotřeby kryptografického modulu ze zdroje napájení. Nemusí se měřit přímo proud, ale i spotřeba napětí nebo výkonu. Z toho důvodu se mu také říká výkonový nebo napěťový. Je dokázáno, že průběh proudové spotřeby není s časem konstantní, ale mění se v závislosti na zpracovávaných operandech a na posloupnosti operací. Na proudové spotřebě se výrazně projevují kryptografické algoritmy, které pracují ve smyčkách a operace opakují. Změny ve spotřebě proudu vznikají už na úrovni elementárních součástek, u procesorů jsou to většinou tranzistory.

Elektromagnetický postranní kanál vzniká podobným způsobem jako proudový postranní kanál. Vyzářená energie elektromagnetického záření odpovídá probíhajícím operacím a operandům zpracovávaným uvnitř modulu.

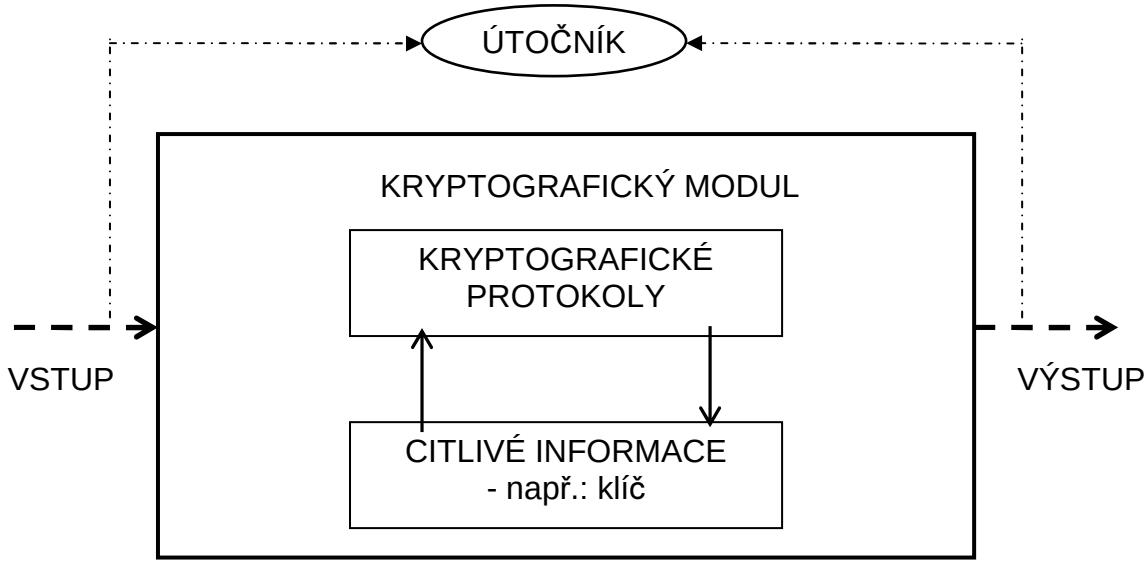
Chybový postranní kanál je založený na chybových hlášeních a systémových selháních, kdy kryptografický modul musí komunikovat s okolím. V běžném provozu jsou tato hlášení nutná pro správnou funkci systému. V dalším případě může útočník tento stav vyvolat uměle, kdy postupným opakováním chybných požadavků dojde ke zjištění některých citlivých informací [2, 8, 9, 13].

## 1.4 ANALÝZA POSTRANNÍHO KANÁLU

Způsob, kterým citlivé informace z kryptografického modulu unikají, je pro každý postranní kanál jiný. V rámci útoku je potom nutné tyto informace zpracovat a vyhodnotit, v kryptografii se tomu říká analýza kanálu (Obr. 1.2). Existují dva základní druhy těchto analýz.

Jednoduchá analýza reprezentuje základní způsob zpracování výsledků. Útočník informace, které získal z postranního kanálu, přímo vyhodnocuje a to bez použití speciálních výpočetních metod.

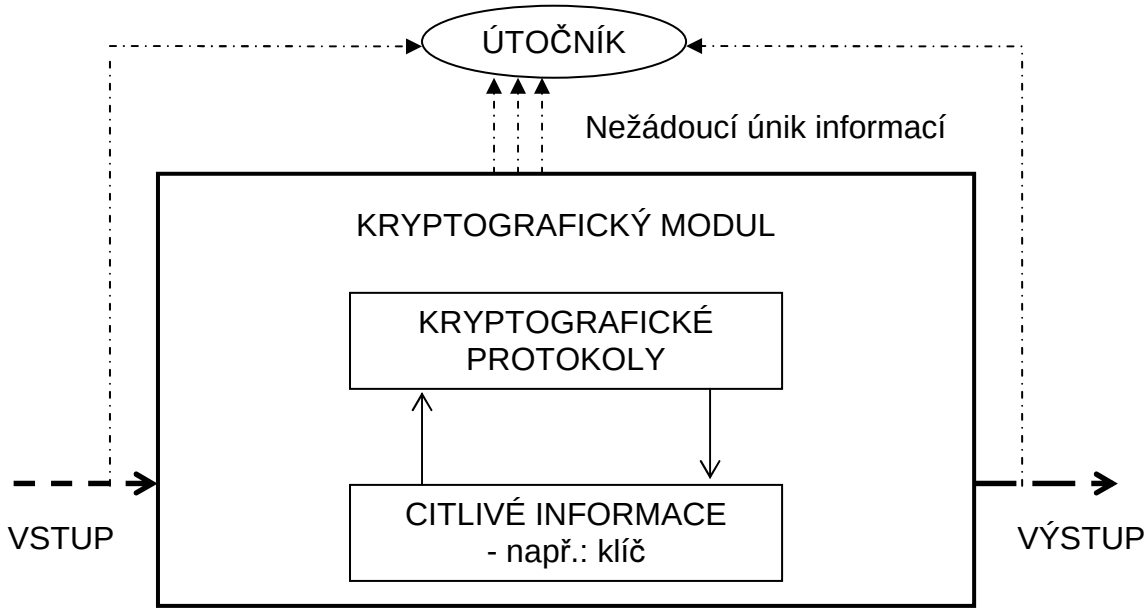
Diferenční analýza je složitější, protože je zapotřebí matematický aparát, ale na druhou stranu umožňuje objevit citlivé informace i z postranních kanálů, kde nejsou předem zřejmé. Také zjednodušuje možnost automatického procesu [2].




Obr. 1.2: Obecná kryptoanalýza.

## 1.5 ÚTOK POSTRANNÍM KANÁLEM

Útok postranním kanálem je napadení konkrétního kryptografického modulu pomocí analýzy vybraného postranního kanálu (Obr. 1.3) [2, 9].




Obr. 1.3: Obecný model útoku na kryptografický modul.

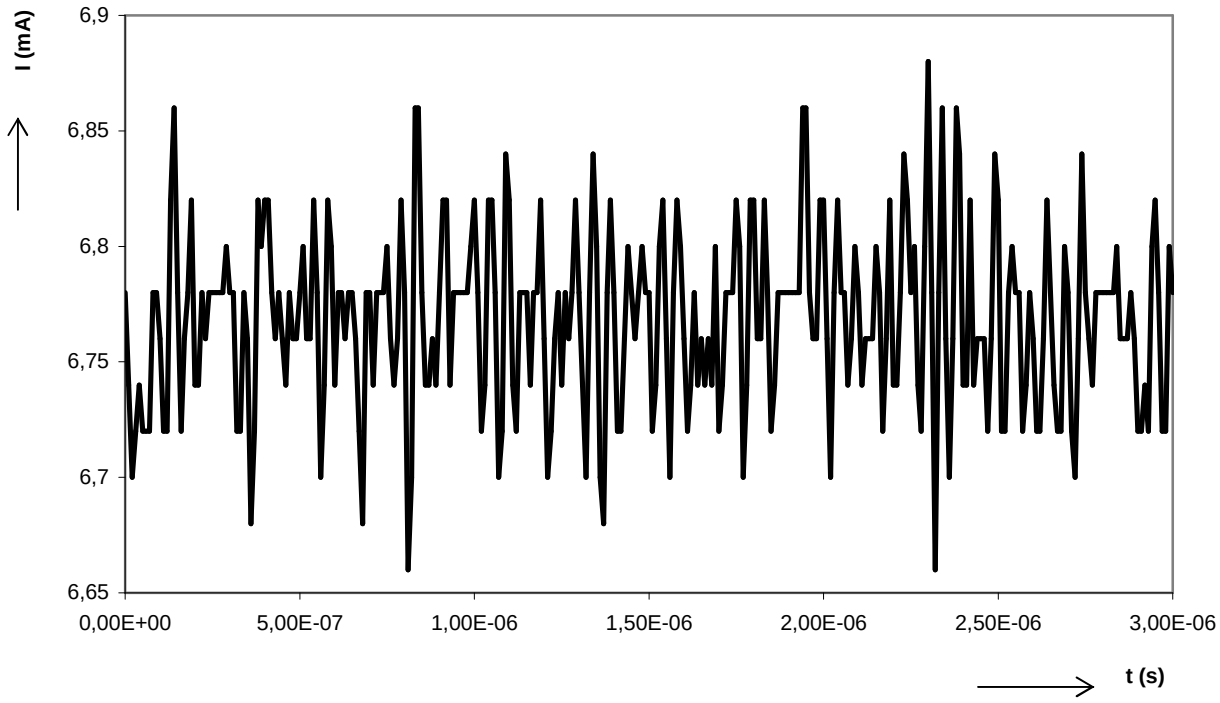
# 2 ÚTOK PROUDOVÝM POSTRANNÍM KANÁLEM

Při útoku proudovým postranním kanálem dochází ke značnému ovlivnění měřeného signálu šumem. Proto se používají speciální metody pro jeho potlačení.

Nejčastěji se tímto typem útoku napadají čipové karty, protože nemají vlastní autonomní zdroj a tak se musí napájet externě. Proto lze bez problémů měřit proudovou spotřebu. Hlavní výhodou útoku postranním kanálem na čipové karty je snadné měření spotřeby při jejich komunikaci se čtecím zařízením [9].

## 2.1 MĚŘENÍ PROUDOVÉ SPOTŘEBY

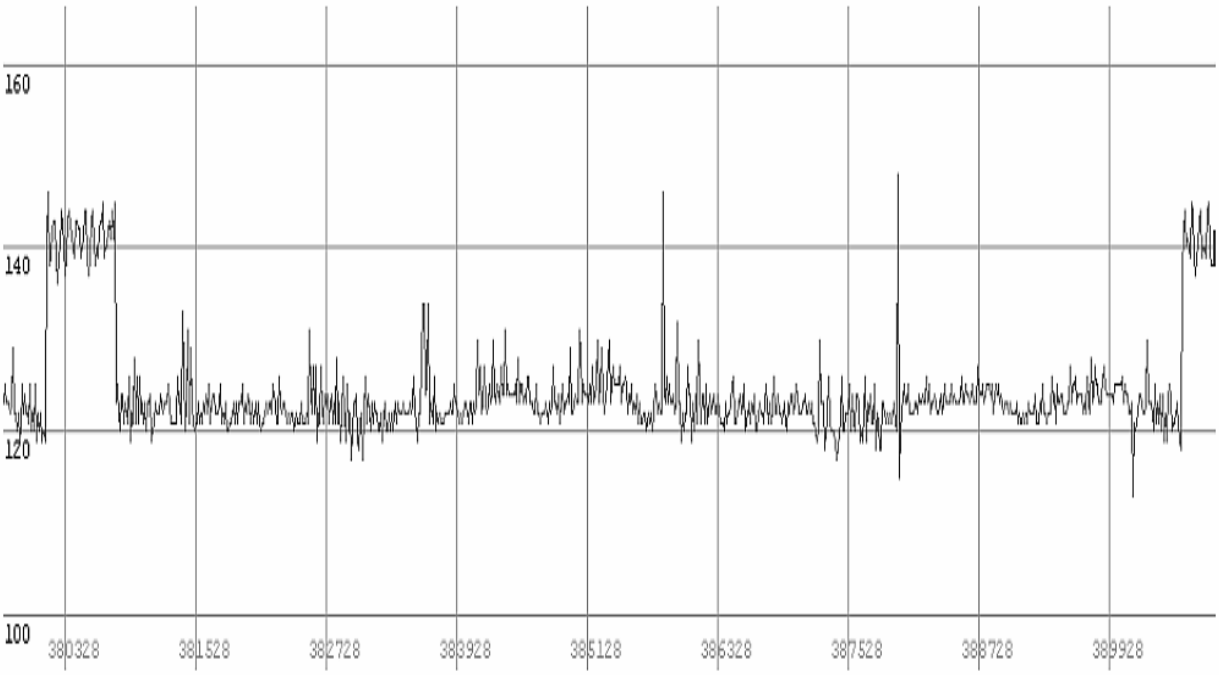
Z měření proudové spotřeby mikroprocesoru (Obr 2.1) je vidět, že se jeho spotřeba s časem mění, tedy že není s časem konstantní.




Obr. 2.1: Proudová spotřeba mikroprocesoru.

Na první pohled se zdá, že zvlněný průběh je pouze šum se stejnosměrnou složkou, ale nahodilost je jen zdánlivá, protože šum se mění se změnami stavů elektronických součástek uvnitř zařízení. Jelikož je těchto součástek mnoho, tak výsledný průběh proudové spotřeby vypadá jako zdánlivě šum. Změny proudové spotřeby vznikají na úrovni základních elektronických součástek, mezi které patří z velké části tranzistory. Probíhající kryptografické operace přímo ovlivňují činnost těchto prvků [2, 8].

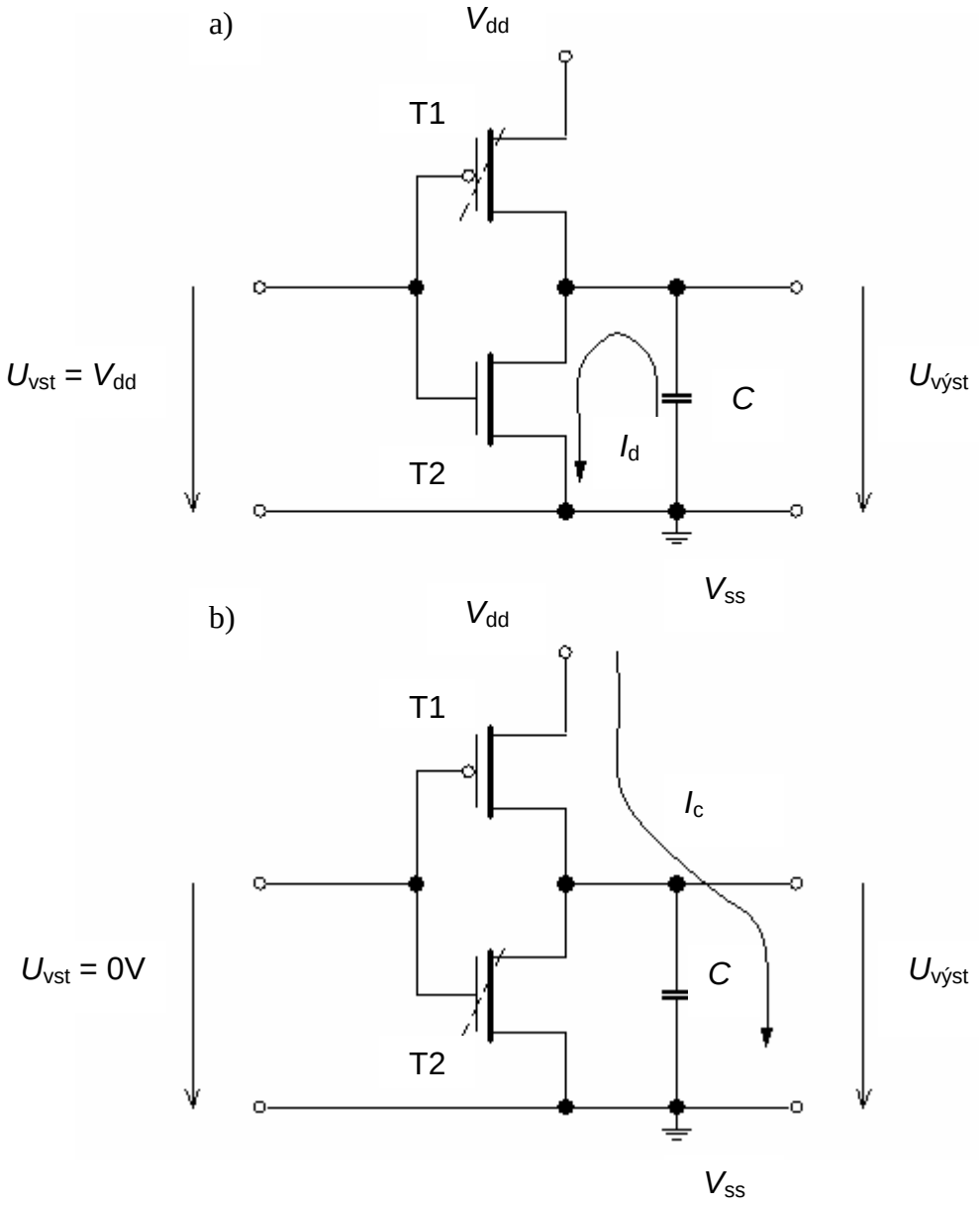
Na obrázku 2.2 je znázorněn průběh první rundy výkonové analýzy algoritmu AES (Advanced Encryption Standard), což je označení pro pokročilý šifrovací standard. Je to nástupce symetrického blokového šifrovacího algoritmu DES (Data Encryption Standard), který byl již před mnoha lety prolomený a není tedy bezpečný. V současné době je sice algoritmus AES nejbezpečnější, protože je momentálně neprolomitelný, ale je jen otázkou času, než se to někomu podaří.




Obr. 2.2: První runda výkonové analýzy algoritmu AES [10].

## 2.2 KONSTRUKCE PROCESORU

Dnes je nejvíce procesorů založeno na technologii CMOS (*Complementary Metal Oxide Semiconductor*). V těchto obvodech je elementárním stavebním prvkem invertor, jehož vnitřní zapojení se skládá ze dvou tranzistorů. Tyto tranzistory jsou zapojeny jako spínače řízené napětím (Obr. 2.3).




Obr. 2.3: Model invertoru CMOS s připojenou interní kapacitní zátěží:

a) Vybíjení ($I_d$), b) Nabíjení ($I_c$).

- V případě, že se na vstup připojí napětí logické úrovně "0", je horní tranzistor otevřen a dolní uzavřen .
- Pokud je na vstup připojeno napětí logické úrovně "1", je dolní tranzistor otevřen a horní uzavřen.

U obou těchto stavů je proudová spotřeba nízká, ale pro každý stav je jiná.

Při přechodu mezi těmito stavy nastává výkonová špička, kdy jsou na okamžik otevřeny oba tranzistory současně a napájení je zkratováno proti zemi. Čemuž se říká dynamická spotřeba. Následně dochází ke vzniku proudové špičky. Její velikost závisí na tom, kolik tranzistorů právě přepíná. Proudovou špičku lze změřit tak, že se do série s $V_{dd}$ nebo $V_{ss}$ zapojí rezistor, na kterém se následně změří úbytek napětí, který odpovídá okamžitému odběru proudu. Tranzistory odebírají malý proud i v klidovém stavu, který se pak mění na teplo či záření. Dominantním zdrojem výkonových změn je nabíjení ($I_c$) a vybíjení ($I_d$) interní kapacitní zátěže, která je připojená na výstupy. Mezi zdroje výkonových změn patří:

- Tepelné vyzařování tranzistorů v klidovém stavu a proudový odběr pro stav logické "0" a pro stav logické "1". Elektrická energie se mění na teplo.
- Proudové špičky při přechodu mezi stavy logické "0" a logické "1".
- Změny proudu při vybíjení a nabíjení parazitní kapacitní zátěže připojené sběrnice při změnách pracovních stavů.

Z toho plyne, že výkonová spotřeba elektronických obvodů přímo závisí na operacích, které v nich probíhají, tedy na množství překlápěných tranzistorů. Analýzou výkonové spotřeby lze zjistit citlivé informace uvnitř kryptografického modulu [2, 8, 13].

## 2.3 VÝKONOVÉ ANALÝZY

Kapitola 2.3 vychází z [2].

Při zpracování programu dochází v kryptografických modulech, které jsou vybaveny mikroprocesory, ke spínání množství tranzistorů. Stavy logické "1", logické "0" a jejich změny závisí přímo na prováděném programu. Tím to způsobem dochází k nežádoucímu zanesení informace o vnitřním stavu programu do výkonové spotřeby.

### 2.3.1 JEDNODUCHÁ VÝKONOVÁ ANALÝZA

Je technika útoku, při které útočník získává informace z postranního výkonového kanálu na základě přímého pozorování průběhu výkonové spotřeby kryptografického modulu.

Kryptografické algoritmy pracují s citlivými informacemi a běží v pravidelných cyklech. Tím ve výkonové spotřebě dochází k pravidelně se opakujícím vzorům. Z kterých se dá na základě obecné znalosti šifrovacích algoritmů spolehlivě stanovit typ konkrétního sledovaného algoritmu a pozorovat jednotlivé fáze průběhu zpracování. Citlivé informace lze pak získat z malých odchylek v opakujících se cyklech.

Tato technika tedy nevyužívá žádné statické metody ani jiné matematické postupy, ale vychází ze znalosti průběhu výkonové spotřeby jednotlivých instrukcí, který je pro každou instrukci typický. Její uskutečnitelnost ale obvykle vyžaduje zkušeného kryptoanalytika.

Jelikož se v různých fázích provádění programu využívají různé instrukce, je výhodné jednoduchou výkonovou analýzou použít hlavně proti kryptografickým protokolům, kde průběh provádění programu závisí na zpracovávaných datech. Značnou nevýhodou je nemožnost analýzu automatizovat.

#### 2.3.1.1 OPATŘENÍ PROTI JEDNODUCHÉ VÝKONOVÉ ANALÝZE

Toto opatření je možné snadno realizovat na softwarové úrovni tak, že se odstraní všechna větvení programu, která závisí na citlivých informacích. Tím se změní charakteristický průběh výkonové spotřeby, který právě jednoduchá výkonová analýza využívá.

Opatření je možné provézt i hardwarově. Tento systém pak provádí vlastní změny ve výkonové spotřebě, takže provedení jednoduché výkonové analýzy je nemožné [2].

### 2.3.2 DIFERENČNÍ VÝKONOVÁ ANALÝZA

Při zpracovávání programu se data pravidelně čtou a zapisují do paměti. Přenáší se po datové sběrnici. Sběrnice má kapacitní vlastnosti, takže dochází k pravidelnému nabíjení a vybíjení jejich cest, což se znovu projevuje ve výkonové spotřebě. Tyto jevy jsou

ještě více nepatrné, proto se při útoku musí použít pravděpodobnostní výpočty a promyšlené matematické postupy.

Diferenční výkonová analýza je technika útoku založena na sestavení hypotetického modelu kryptografického modulu. Tento model musí zahrnovat simulovaný postranní kanál. Ze skutečného a hypotetického modulu se pak pomocí statických metod získané výsledky vyhodnotí.

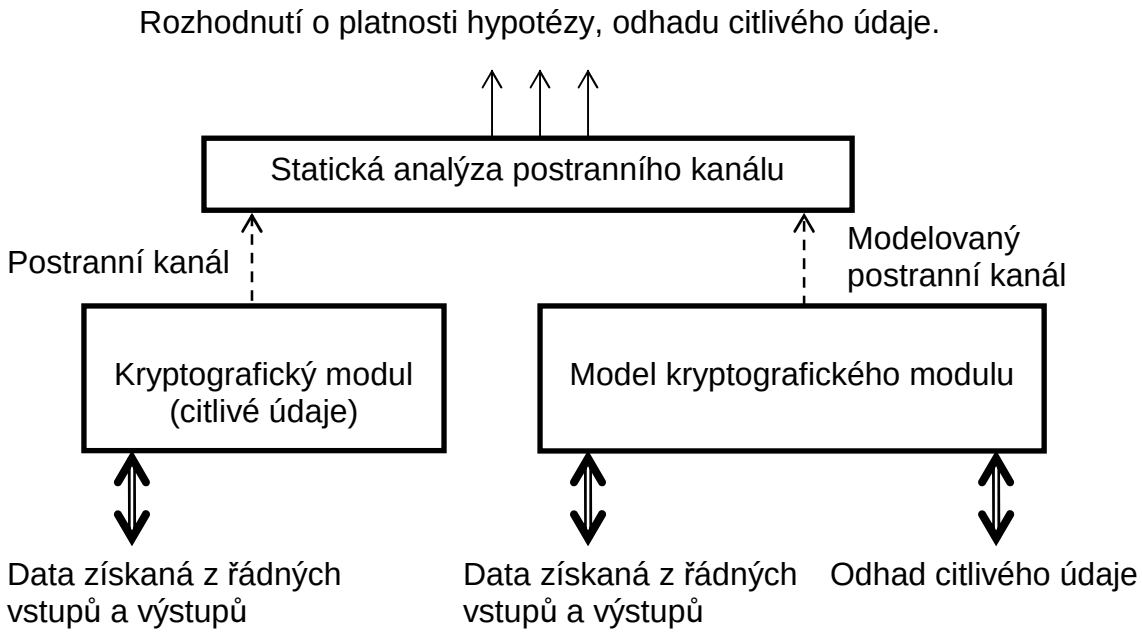
Útok na výkonový postranní kanál touto metodou patří mezi nejnebezpečnější. Sledují se korelace mezi výkonovým průběhem a daty, které program zpracovává. Toto spojení není většinou příliš zřejmé, proto je k jeho nalezení zapotřebí použít různé statické metody.

Na obrázku 2.4 je obecný model diferenční analýzy. Jeho základem je skutečný kryptografický modul. Dále se předpokládá, že data zpracovávaná na reálném i hypotetickém modulu jsou shodná. Korektní odhad citlivého údaje se získá z korelace mezi oběma postranními kanály.

Jednorozměrný postranní kanál je takový postranní kanál, který má pro každý časový okamžik pouze jednu hodnotu a skládá se jen z jednoho typu kanálu.

Vícerozměrný postranní kanál je takový postranní kanál, který má pro každý časový okamžik hned několik hodnot a skládá se z více typů postranních kanálů.

Pokud jsou data získána a analyzována z jednorozměrného postranního kanálu, jedná se o diferenční analýzu prvního řádu. Jestliže jsou získána a analyzována z vícerozměrného postranního kanálu, jde o diferenční analýzu vyšších řádů.




Obr. 2.4: Model diferenční analýzy.

#### 2.3.2.1 HYPOTETICKÝ MODEL SKUTEČNÉHO KRYPTOGRAFICKÉ MODULU

Pro úspěšný útok provedený pomocí diferenční výkonové analýzy je nutné vytvořit tento hypotetický model, který vychází ze znalosti skutečného kryptografického modulu s výkonovým postranním kanálem. Slouží pro co možná nejpřesnější odhady výstupů z postranního kanálu skutečného modulu. K jeho realizaci je nezbytné využít všechny možné informace o reálném modulu. Kvalita tohoto modulu se posuzuje podle počtu výstupních hodnot, které je schopen pro jeden postranní kanál produkovat nebo podle počtu hodnot, které může z postranního kanálu odhadovat.

Útočníka, který využívá pro napadení postranního kanálu hypotetický model, je možné zařadit do jedné z následujících skupin.

- Nemá podrobné informace o reálném kryptografickém modulu. Má pouze základní znalosti, které zahrnují například vědomosti o jeho algoritmech nebo o blokovém upořádání jeho hardwarových částí. Není schopen tedy sestavit kvalitní model.

- Má detailní znalosti o reálném kryptografickém modulu. Má spolehlivé informace o jeho výkonových charakteristikách a je schopen si opatřit matematický model modulu. Může tedy sestavit kvalitní a výstižný model.

Obecně však útočník nepatří do žádné z těchto skupin, ale pohybuje se někde mezi nimi.

#### 2.3.2.2 VÝŠŠÍ EFEKTIVITA DIFERENČNÍ VÝKONOVÉ ANALÝZY

Přídavný šum, který naměřené výkonové spotřeby obsahují, zvyšuje minimální počet měření těchto průběhů. Což je důležité, jinak by nebyl útok proveditelný.

Tento šum vzniká z mnoha důvodů. Může jít například o tepelný šum, nebo o kvantovací chybu způsobenou odlišností hodinového signálu použitého ke vzorkování a signálu uvnitř kryptografického modulu. V neposlední řadě může vznikat vlivem elektromagnetického záření v blízkosti měřícího stanoviště.

Značné množství šumu pak v jednotlivých naměřených průbězích způsobí časové odchylky. Proto je pro úspěšný útok nutné měření několikrát opakovat. Lepším řešením ale je měření zefektivnit. Počet potřebných měření lze snížit potlačením zdrojů šumu či automatizací procesu, což v případě algoritmu DES představuje desítky měření. Automatizace procesu může vézt i k prolomení některých protiopatření kryptografického modulu.

#### 2.3.2.3 STATISTICKÉ METODY PRO DIFERENČNÍ ANALÝZU

Princip diferenční analýzy spočívá v porovnání výstupu ze skutečného kryptografického modulu a jeho hypotetického modulu. Toto porovnání je realizováno pomocí statistických metod. Zmíněné výstupy pak představují statistické hodnoty. Ty je pak možné s využitím metod pro vyhodnocování statistických hodnot porovnat. Mezi tyto metody patří například statistická metoda založená na rozdílu středních hodnot a statistická metoda založená na korelačním koeficientu.

- **Statistická metoda založená na korelačním koeficientu.**

Statistická metoda založená na korelačním koeficientu slouží pro určení lineární závislosti mezi dvěma náhodnými proměnnými. Příkladem je Pearsonův korelační koeficient definovaný jako [2]:

$$\rho(X,Y) = \frac{\mathrm{cov}(X,Y)}{\rho_X.\rho_Y} = \frac{E((X-\mu_X)(Y-\mu_Y))}{\rho_X.\rho_Y}. \tag{2.1}$$

Korelační koeficient se pohybuje v rozmezí hodnot -1 až 1. Hodnota -1 korelačního koeficientu značí nepřímou závislost. To znamená, že změna v jedné skupině způsobí opačnou změnu ve skupině druhé. Hodnota 0 korelačního koeficientu značí, že mezi hodnotami obou skupin neexistuje žádná statisticky zjistitelná závislost. Hodnota 1 korelačního koeficientu značí přímou závislost, ta představuje dokonalou korelaci mezi hodnotami obou skupin.

Korelační koeficient je využitý například při hledání Hammingovy váhy zpracovávané proměnné za využití postranního kanálu u šifrovacího algoritmu DES. Reálný koeficient v tomto případě je [2]:

$$r(I_i(t),D(X_i,K_h)) = \frac{\sum_{i=1}^{M} I_i(t).D(X_i,K_{h)}}{\sqrt{\sum_{i=1}^{M}(I_i(t)-\overline{I_i(t)})^2.\sum_{i=1}^{M}(D(X_i,K_h)-\overline{D(X_i,K_h)})^2}} - \frac{\frac{1}{M}\sum_{i=1}^{M} I_i(t).\sum_{i=1}^{M} D(X_i,K_h)}{\sqrt{\sum_{i=1}^{M}(I_i(t)-\overline{I_i(t)})^2.\sum_{i=1}^{M}(D(X_i,K_h)-\overline{D(X_i,K_h)})^2}}, \tag{2.2}$$

kde $D(X_i,K_h)$ představuje Hammingovu váhu čtyř bitového výstupu daného S-boxu v první rundě pro známí otevřený text $X_i$ a odhadovaný klíč $K_h$. Průběh výkonové spotřeby je označen $I_i(t)$. Rovnicí (2.2) stanovený reálný korelační koeficient $r(I_i(t),D(X_i,K_h))$ konverguje k teoretickému korelačnímu koeficientu $\rho(I_i(t),D(X_i,K_h))$ s rostoucím počtem měření [2]:

$$\rho(I_i(t),D(X_i,K_h)) = \lim_{M\to\infty} r(I_i(t),D(X_i,K_h)). \quad (2.3)$$

Množství potřebných měření k úspěšně uskutečnitelnému útoku lze statisticky určit pomocí Fischerovy Z-transformace teoretického korelačního koeficientu. Tím se stanoví nezbytný počet měření, který zajistí, že nesprávně odhadnutý klíč test nebude testem označen za správný. Předpoklad pro určení množství měření je opakování útoku postranním kanálem za stále stejných podmínek. Analýzou hodnot reálných korelačních koeficientů $r(I_i(t),D(X_i,K_h))$ při neměnném odhadnutém klíči je možné zjistit, jestli jsou jeho hodnoty normálního rozložení a jejich průměr konverguje k teoretickému korelačnímu koeficientu, a nebo nejsou hodnoty korelačního koeficientu normálního rozložení. Proto je důležité nejprve pomocí Fischerovy Z-transformace hodnoty reálného korelačního koeficientu transformovat do hodnot normálně rozložených [2]:

$$z = \frac{1}{2}.\left(\frac{1+r}{1-r}\right), \quad (2.4)$$

$$\mu_z = \frac{1}{2}\left(\frac{1+\rho}{1-\rho}\right), \quad (2.5)$$

$$\sigma_z^2 = \frac{1}{M-3} \approx \frac{1}{M}; M >> 1. \quad (2.6)$$

Střední hodnoty korelačního koeficientu pro správně a nesprávně odhadnutý klíč jsou [2]:

$$\mu_{z,K=K_h} = \frac{1}{2}.\left(\frac{1+\rho}{1-\rho}\right), \quad (2.7)$$

$$\mu_{z,K\neq K_h} = \frac{1}{2}.\left(\frac{1+0}{1-0}\right), = 0. \quad (2.8)$$

Z výše zmíněných informací lze říci, že je větší pravděpodobnost, že výsledek transformovaného reálného korelačního koeficientu bude pro správný odhad klíče $z_{K=K_h}$, než pravděpodobnost, že výsledek transformovaného reálného korelačního koeficientu bude pro správný odhad klíče $z_{K\neq K_h}$. Dále je nutné určit odstup $d = z_{K=K_h} - z_{K\neq K_h}$, kdy odstup $d$ je

normální distribuce se střední hodnotou $\mu_d = \frac{1}{2}.\left(\frac{1+\rho}{1-\rho}\right)$ a variancí $\sigma_d^2 = \frac{2}{M-2}$. Tyto dva fakty je možné shrnout do rovnice pro pravděpodobnosti odhadu klíče [2]:

$$P(z_{K=K_h} > z_{K\neq K_h}) = P(d = z_{K=K_h} - z_{K\neq K_h} > 0),$$
$$P(z_{K=K_h} > z_{K\neq K_h}) = 1 - P(d = z_{K=K_h} - z_{K\neq K_h} < 0), \qquad (2.9)$$
$$P(z_{K=K_h} > z_{K\neq K_h}) = \Phi\left(\frac{\frac{1}{2}.\ln\left(\frac{1+\rho}{1-\rho}\right)}{\sqrt{\frac{2}{M-3}}}\right).$$

Řešením rovnice je možné získat počet potřebných měření [2]:

$$M = 3 + 8.\left(\frac{\Phi^{-1}(P(d>0}{\ln\left(\frac{1+\rho}{1-\rho}\right)}\right)^2. \qquad (2.10)$$

Reálné hodnoty jsou $P(d>0) = 0{,}9999$ případně $\Phi^{-1}P(d>0) \approx 3{,}719$. Což ve výsledku znamená, že 99,99% všech korelačních koeficientů by mělo být menších než korelační koeficienty pro správně odhadnutý klíč.

- **Statistická metoda založená na rozdílu středních hodnot.**

Statistická metoda založená na rozdílu středních hodnot slouží pro porovnání dvou skupin naměřených hodnot. To se děje výpočtem rozdílu středních hodnot těchto skupin. Postup:

1. naměří se několik řad hodnot,
2. tyto řady se rozdělí do dvou skupin,
3. z každé skupiny se získá jedna řada jako střední hodnota všech řad,
4. vypočte se rozdíl mezi středními průběhy obou skupin.

Pokud jsou střední hodnoty od sebe vzdáleny a liší se tedy o definovanou odchylku, jsou tyto dvě skupiny považovány za rozdílné. Jinak jsou považovány za shodné.

Na této snadné metodě je založený i tzv. Kocherův útok, viz [2]. Model zde použitý je dost slabý, obecný a nezahrnuje žádné informace o fyzické implementaci. Umožňuje analyzovat pouze jeden bit pro danou rundu. Ze znalostí o implementaci je možné tento model vylepšit tak, aby byl schopný odhadnout větší počet bitů, tím se pak docílí toho, že třídění naměřených průběhů do zmíněných dvou skupin je daleko přesnější.

Tento způsob útoku je poměrně jednoduchý a je ideální pro pochopení diferenční výkonové analýzy. Má ale i množství nedostatků, tím hlavním je nemožnost zpracovávat více než dvě skupiny současně. Takže odhad lze testovat pouze pro binární hodnoty. Pokud model kryptografického modulu umožňuje přesně stanovit odhad pro vícero bitů najednou, je zapotřebí využít jinou metodu.

Metoda, která nahrazuje statistickou analýzu založenou na rozdílu středních hodnot se nazývá analýza rozptylu ANOVA (Analysis of variance).

# 3 ROZDĚLENÍ MIKROPROCESORŮ

Mikroprocesory se dělí podle instrukční sady do dvou hlavních skupin [4], [12], [15]:

- **CISC (Complex Instruction Set Computer):**

Procesory s komplexní instrukční sadou. Pracují se složitými instrukcemi, jejichž provedení trvá různou dobu, i několik strojových cyklů. Výzkumem se došlo k závěru, že 80% kódu programu využívá jen 20% těchto instrukcí, což vedlo ke vzniku další skupiny.

- **RISC (Reduced Instruction Set Computer):**

Procesory s redukovanou instrukční sadou. U tohoto typu procesorů se instrukce, které jsou složité a jen zřídka používané, vypisují v kódu programu a nezabírají tak zbytečně prostor v řídící paměti čipu. Instrukční sada obsahuje jen jednoduché a nejčastěji používané instrukce.

Instrukce mají jednu pevně danou délku a pevný formát, který jednotlivým bitům nebo skupinám bitů přesně stanovuje funkci. Program sice obsahuje větší počet instrukcí, ale instrukci provede mnohem rychleji.

Řídící obvody u architektury RISC zabírají na čipu pouze 6-10% místa a vykonání instrukce trvá jeden strojový cyklus, vyjímkou je jen komunikace s pamětí.

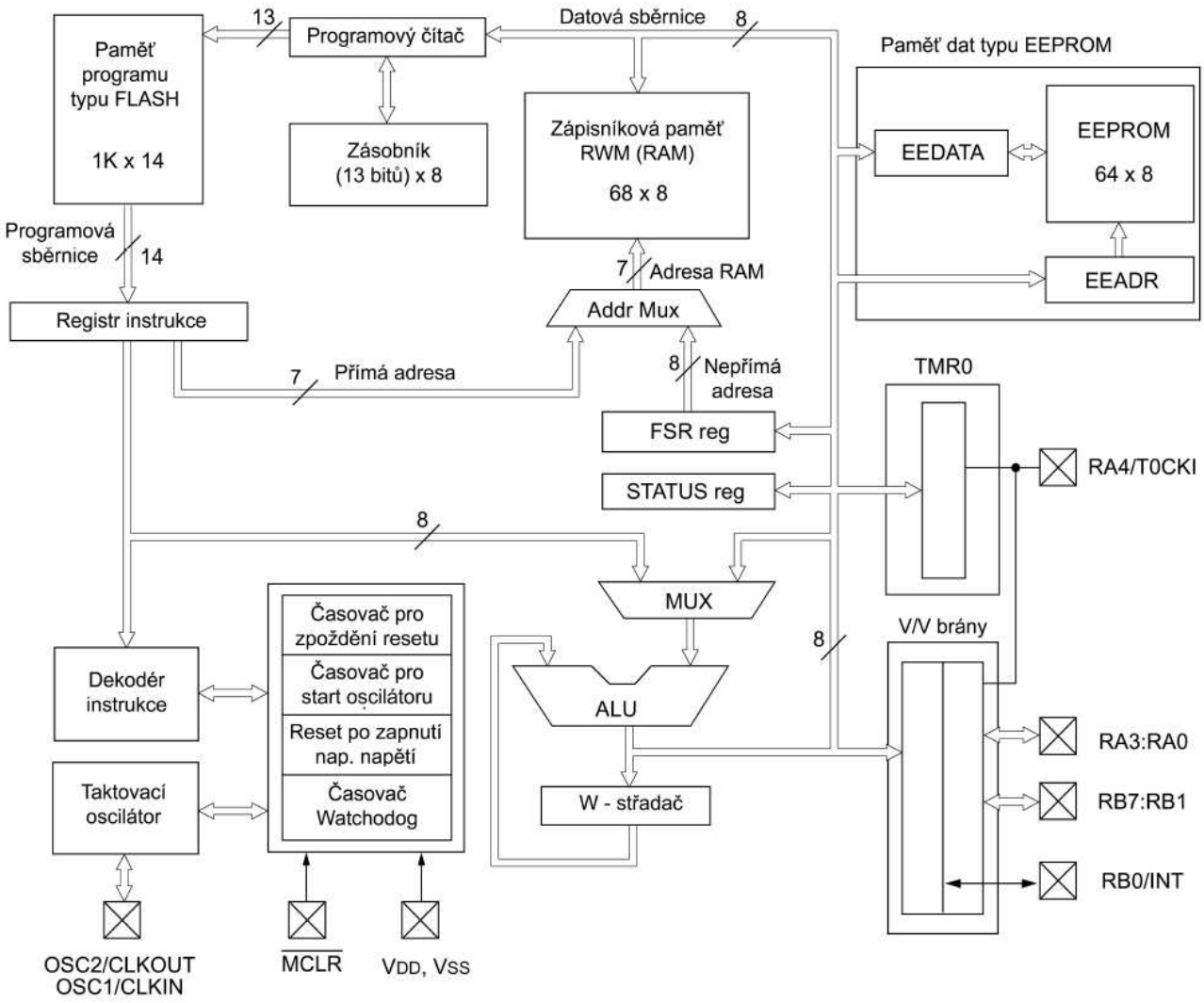
Aby se dosáhlo co nejvyšší rychlosti, kterou snižují právě přístupy do paměti, využije se volné místo na čipu pro soubory registrů, ke kterým je jednocyklový přístup. Žádná instrukce neadresuje více něž jedno paměťové místo. Registry jsou víceúčelové, takže instrukce může využít každý z nich. S pamětí procesor komunikuje po sběrnici. Pro spolupráci s hlavní pamětí se používají pouze dvě instrukce, a to zápis do paměti a čtení z paměti. Tyto typy procesorů umožňují také pipelining neboli řetězení instrukcí.

# 4 PIC16F84A

Procesor PIC16F84A od firmy Microchip Technology Inc. je rychlejší verzí procesoru PIC16F84, která obsahuje drobná vylepšení elektrických vlastností. Je založený na technologii CMOS a pracuje na architektuře RISC. I když je to dnes už poměrně starší typ procesoru, má široké spektrum využití, používá se jak ve spotřební elektronice - mikrovlnné trouby, dálkové ovladače, televizní přijímače, tak i v automatizační technice - regulátory, inteligentní snímače, smart karty [1, 11].

## 4.1 JÁDRO PROCESORU PIC16F84A

Mikroprocesor 16F84A je osmibitový flash/EEPROM jednočipový mikropočítač s 18 vývody postavený na Harvardské architektuře, která odděluje paměť programu a paměť dat (Obr. 4.1).




Obr. 4.1: Blokové schéma procesoru PIC16F84A [11].

## 4.2 INSTRUKCE PIC16F84A

Procesor obsahuje 35 jednoduchých instrukcí, ty se vykonají během jednoho instrukčního cyklu. Vyjímkou jsou jen instrukce, u kterých dochází ke větveni programu, ty se vykonají během dvou instrukčních cyklů. Jeden instrukční cyklus trvá 4 takty. Výpočet doby vykonání jednoho instrukčního cyklu pro taktovací frekvenci 4 MHz:

$$t = \frac{1}{f} \cdot 4 = \frac{1}{4.10^6}.4 = 1\ (\mu s), \tag{4.1}$$

výpočet doby vykonání jednoho instrukčního cyklu pro taktovací frekvenci 20 MHz:

$$t = \frac{1}{f} \cdot 4 = \frac{1}{20.10^6}.4 = 0{,}2\,(\mu s), \tag{4.2}$$

kde u obou výpočtů (4.1, 4.2) je *f* taktovací frekvence oscilátoru a *t* je doba vykonání jednoho strojového cyklu [3, 11].

Během instrukčního cyklu se provedou následující operace [6]:

1. načtení instrukce (fetch),
2. rozpoznání instrukce (decode),
3. vykonání instrukce (execute),
4. uložení výsledku (store).

Dochází tedy k rozložení úlohy na menší části.

Procesor PIC16F84A má tři druhy instrukcí [1, 11]:

- bytově orientované instrukce (např.: MOVF, NOP, CLRW),
- bitově orientované instrukce (BSF, BCF, BTFSS, BTFSC),
- řídící instrukce (např.: GOTO, MOVLW, CALL).

A ty podle vykonávané funkce lze rozdělit do skupin:

- instrukce nulování a nastavení (např.: BSF, CLRW),
- instrukce pro přesuny dat (např.: MOVLW, MOVF),
- instrukce pro práci s podprogramy a přerušením (např.: CALL, RETURN),
- instrukce provádějící aritmetické a logické operace (např.: ANDLW, ADDLW),
- instrukce skoků v programu (např.: GOTO, BTFSS),
- zvláštní instrukce (např.: NOP, TRIS),
- direktivy assembleru (např.: ORG, END, EQU).

# 5 MĚŘENÍ ODBĚRU PROUDU

Následující kapitola vychází z [8].

Princip výkonové analýzy je založený na sledování proudové spotřeby elektrického zařízení. Kdy, jak je popsáno v kapitole 2, dochází při různých operacích k různému počtu překlopení tranzistorů a tedy i k různému odběru proudu. Proudovou spotřebu je možné měřit za předpokladu, že je mikroprocesor napájen konstantním napětím z ideálního zdroje napětí. Výkonová spotřeba je přímo úměrná spotřebě proudu:

$$p(t) = U.i(t), \tag{5.1}$$

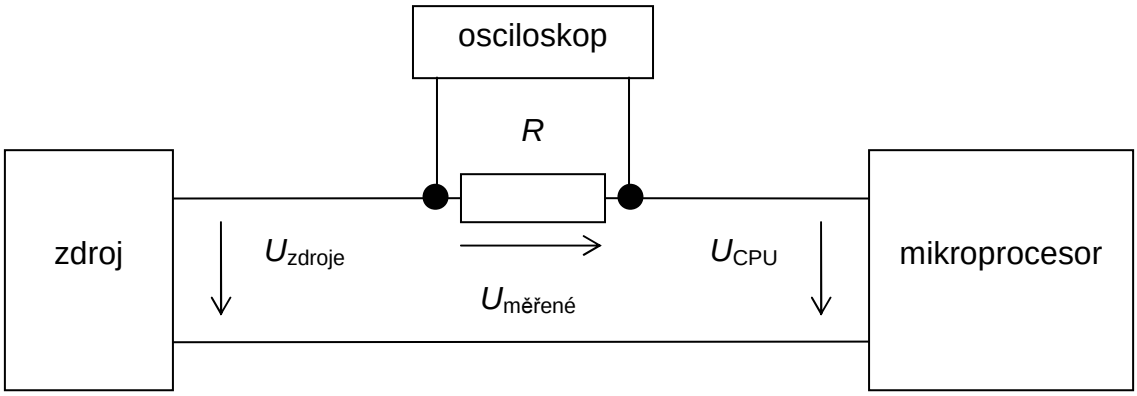
kde $p(t)$ je příkon v čase $t$, $U$ je napětí zdroje a $i(t)$ je proud v čase $t$.

## 5.1 ODPOROVÝ BOČNÍK

Výhodou odporového bočníku (Obr. 5.1) je jeho frekvenční nezávislost. Mezi zdroj a napájení mikroprocesoru je zapojen malý rezistor, v této práci je to rezistor o velikosti 1 Ω. Tím se docílí toho, že naměřené napětí na bočníku přímo odpovídá odebíranému proudu, což plyne z Ohmova zákona:

$$I = \frac{U}{R} \rightarrow U = I.R = I.1 \Rightarrow U = I, \tag{5.2}$$

kde $I$ je elektrický proud, $U$ je elektrické napětí a $R$ je elektrický odpor.




Obr. 5.1: Jednoduché blokové schéma zapojení.

Naopak, pokud se vybere nevhodný rezistor s velkým odporem, dojde k nesprávné funkci mikroprocesoru nebo k jeho poškození, které způsobí napětí nahromaděné v kapacitní zátěži. Osciloskop se do obvodu připojí pomocí diferenční nebo pasivní sondy.

## 5.2 ZDROJE ŠUMU

V elektrických obvodech se šum objevuje běžně. Při analýze postranních kanálů je měřené napětí malé a šum je rušivým, nežádoucím elementem, který měření znehodnocuje, proto ho je nutné potlačit.

Interní šum vzniká v měřeném objektu na všech odporech vlivem vlastního pohybu nosičů náboje. Pohyb nosičů je závislý na teplotě. Tento šum nelze odstranit.

Externí šum vzniká v externím zdroji rušení a do měřeného objektu je přenesen elektromagneticky nebo napájecími vodiči. Šum lze potlačit vstupními filtry, stíněním obvodu od okolí a správným použitím měřícího zařízení.

Kvantizační šum vzniká na A/D převodnících a je dán použitým A/D převodníkem. Lze ho potlačit vícebitovým A/D převodníkem.

Algoritmický šum vzniká v mikroprocesoru vlivem změny obsahu registrů. Potlačit ho lze tak, že mikroprocesor bude vykonávat stále stejnou instrukci se stejnými daty.

Většinu šumů je možné při měření snížit průměrování. Výsledný průběh odběru proudu je pak průměrem několika opakovaných měření stejných instrukcí se stejnými daty. Avšak šum úplně odstranit nelze.

# 6 PRAKTICKÁ REALIZACE

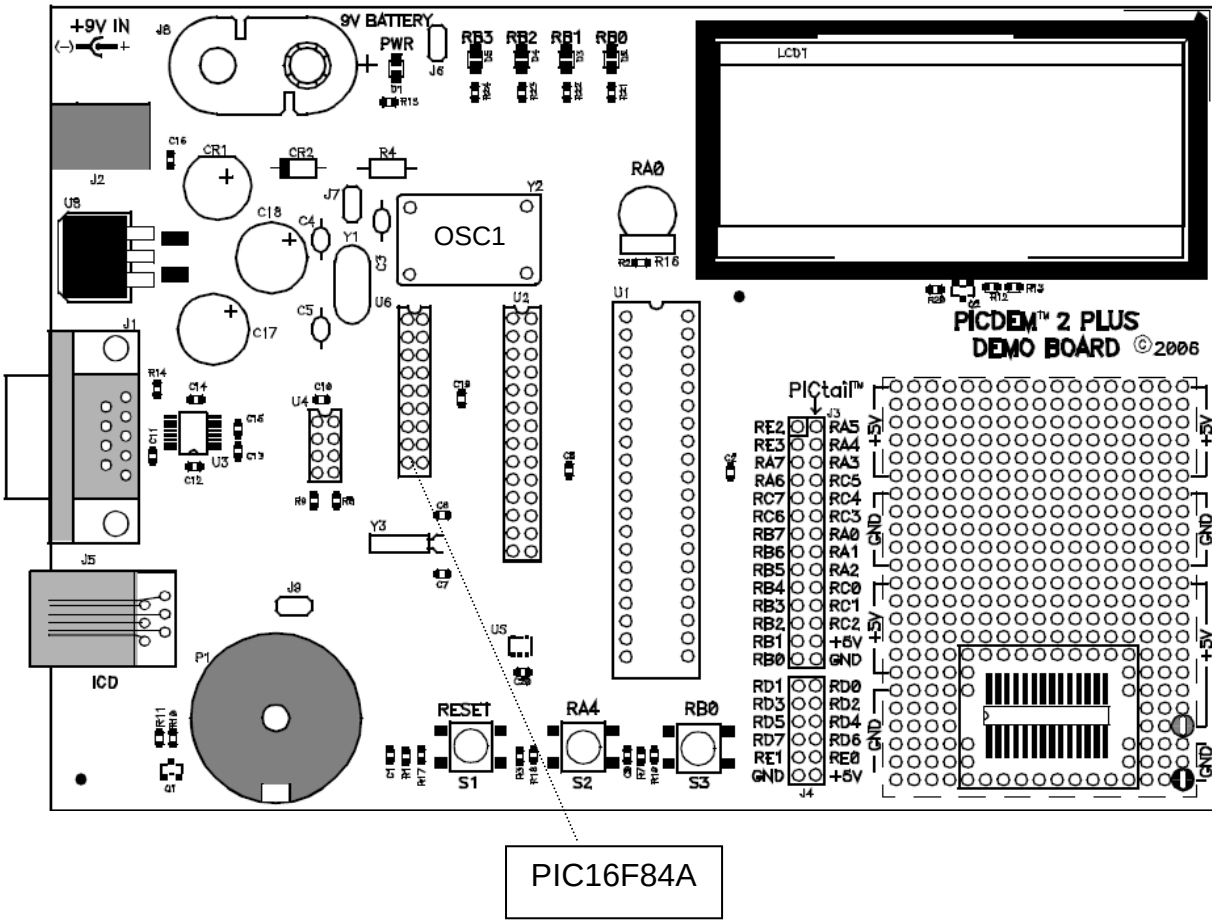
## 6.1 POUŽITÉ PŘÍSTROJE A SOUČÁSTI

Pro analýzu proudového postranního kanálu procesoru PIC16F84A a všechna experimentální měření, která byla nutná pro správný odečet napětí, které mikroprocesor odebírá při zpracování jednotlivých instrukcí, byly použity následující přístroje a součásti:

- Přípravek - Microchip, PICDEM$^{TM}$ 2 PLUS, demo board 2006.
- Vývojové prostředí MPLAB IDE v. 8.33.
- Digitální osciloskop - Agilent DSO3102A.
- Digitální osciloskop (bateriový) - Gwinstek 2204
- Diferenční sonda - Agilent N2772A.
- Pasivní napěťová sonda - Tek P6139A.
- Pasivní sonda - Agilent N2862A.
- Oddělovací transformátor Diametral
- Oscilátor - JWT, 4 MHz, 5V.
- Oscilátor - 20 MHz, 5V
- Rezistor - 1 Ohm.
- Mikroprocesor PIC16F84A.

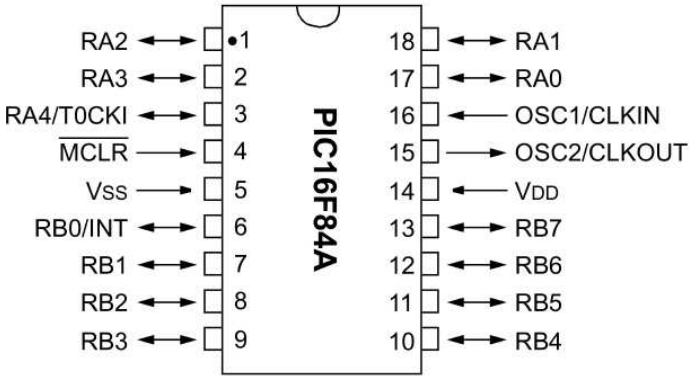
## 6.2 PICDEM$^{TM}$ 2 PLUS

Na obrázku 6.1 je nákres zkušební desky s vyznačením místa pro osazení mikroprocesoru PIC16F84A. Obrázek 6.3 pak představuje názorné schéma zapojení přípravku obsahující patici pro osmnácti pinový mikroprocesor a oscilátor. Tato deska umožňuje tři různé způsoby napájení. Pro následující měření však bylo použito napájení 5 V stejnosměrných, dále byl použitý externí oscilátor OSC1 označený též jako Y2.

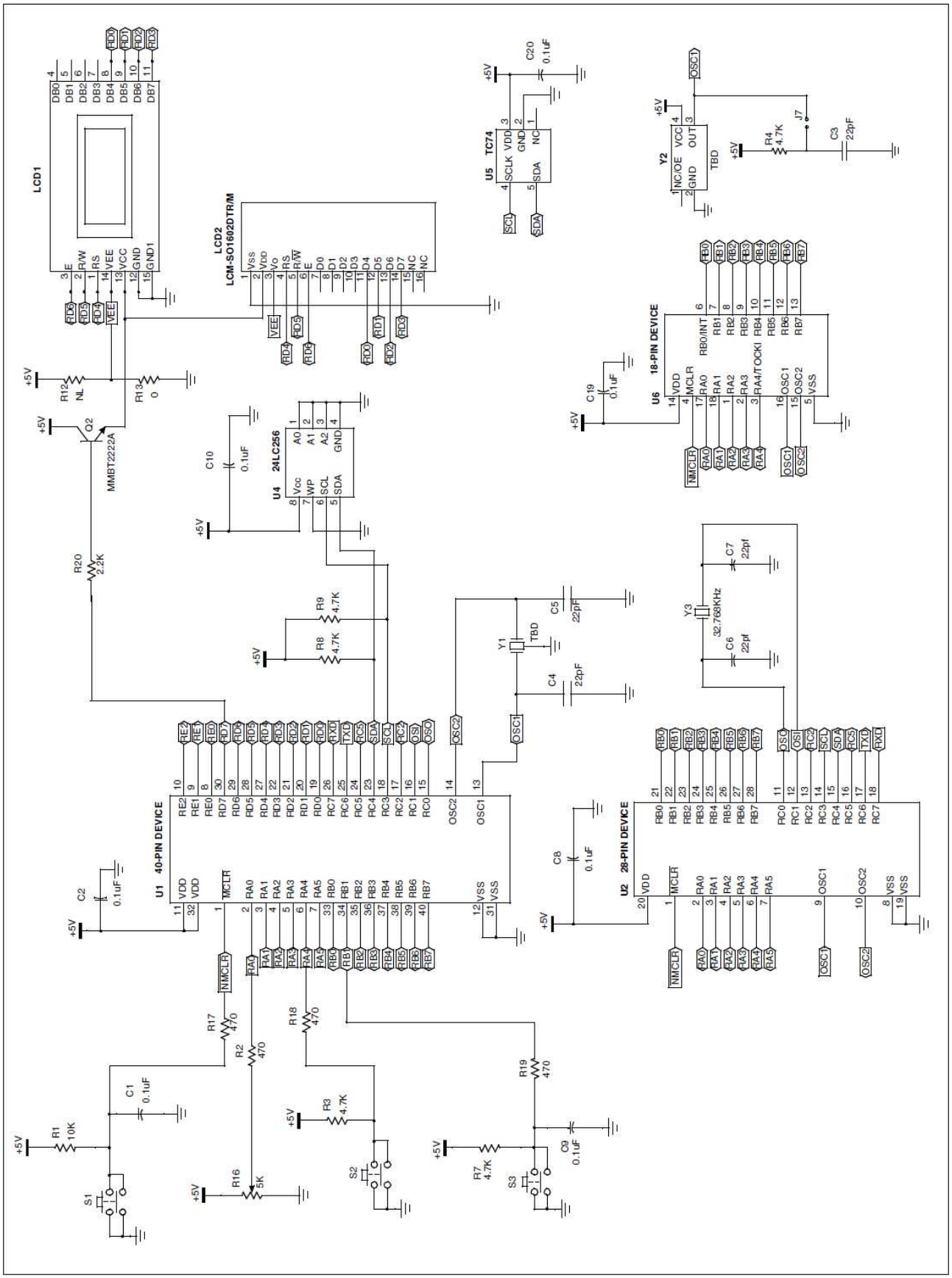



Obr. 6.1: Rozložení přípravku PICDEM$^{TM}$ 2 PLUS [7].

Rezistor o velikosti 1 Ω byl zapojen do série mezi kladné napájecí napětí $V_{DD}$ procesoru PIC16F84A a 14. pin patice v přípravku, který je určený právě pro toto napětí (Obr. 6.2). Na rezistoru se poté pomocí osciloskopu odečítalo napětí při vykonávaní jednotlivých programů. To pak přímo odpovídalo velikosti proudu.




Obr. 6.2: Rozložení pinů procesoru PIC16F84A [11].




Obr. 6.3: Schéma zapojení přípravku PICDEM™ 2 PLUS [7].

## 6.3 ANALYZOVANÉ PROGRAMY

Tato část obsahuje programy naprogramované v assembleru, neboli jazyku symbolických adres. Tyto programy byly vytvořeny ve vývojovém prostředí MPLAB IDE v. 8.33, přes které byly také pomocí datového kabelu postupně nahrány do zkoumaného procesoru PIC16F84A, který již byl v přípravku osazený.

Celkem bylo vytvořeno devět programů. Ty se jmenují podle instrukce, kterou provádí a jsou zpracovávány cyklicky, tedy v nekonečné smyčce. Šest z nich pak bylo použito pro experimentální měření. Mezi tyto programy patří programy ADD, AND, SWAP, XOR, NOP a INC. Programy, které pro měření nebyly použity jsou SUB, IOR a RR.

Protože jsou programy účelně vytvořeny tak, že se prakticky liší pouze v tom řádku, který určuje, jaká instrukce se bude provádět, je zde pro příklad uvedena pouze ukázka hlavní části programu XOR, který provádí operaci exkluzivního logického součtu. Za středníkem je pak uveden komentář, co se v daném kroku provádí. U ostatní programů jsou uvedeny pouze názvy a popis instrukce, kterou realizují.

Pouze u programu AND je binární hodnota vstupních dat 10011001. Zatímco u ostatních programů je binární hodnota vstupních dat 01010101. A to z důvodu porovnání vlivu vstupních dat na výsledný průběh.

Všechny programy jsou pak uvedeny v elektronické příloze.

- **Program XOR.**

```
Main
        movlw B'01010101'; načtení počátečních uživatelských dat
        movwf uziv; přesunutí obsahu pracovního registru do registru uziv
        movlw B'11111111'; načtení nových dat do pracovního registru
Xor     bsf LED2; synchronizační signál ve stavu logická "1" (1 instrukční cyklus)
        xorwf uziv,f; provede XOR mezi registrem W a registrem uziv (1 instrukční cyklus)
        bcf LED2; synchronizační signál ve stavu logická "0" (1 instrukční cyklus)
        goto Xor; větvení programu a nepodmíněný skok na adresu Xor (2 instrukční cykly)
        end; konec programu
```

Činnost programu je blíže vysvětlena v kapitole 6.7.

- **Program ADD.**

Instrukce ADD sečte obsah registru W s registrem uziv.

- **Program AND.**

Instrukce AND provede logický součin mezi registrem W a registrem uziv.

- **Program INC.**

Instrukce INC zvětší obsah registru uziv o jedničku.

- **Program NOP.**

Instrukce NOP neprovede žádnou operaci.

- **Program SWAP.**

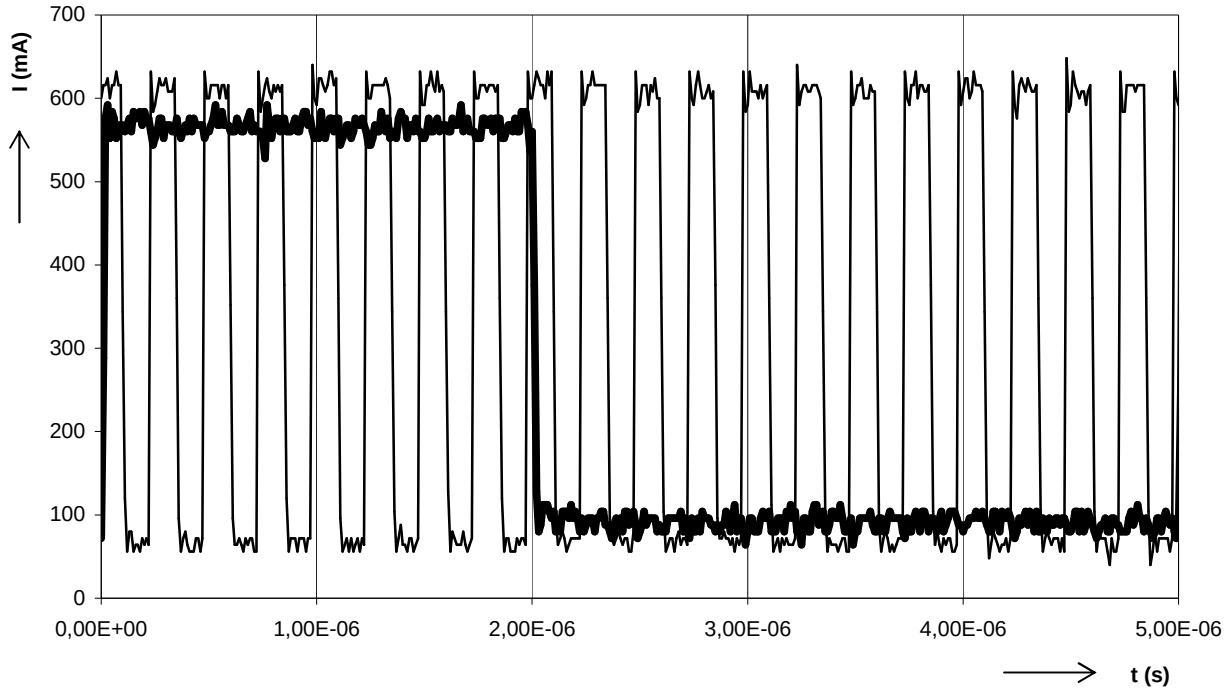
Instrukce SWAP prohodí horní a dolní půlbyte registru uziv.

## 6.4 POSTUP PŘI ANALÝZE

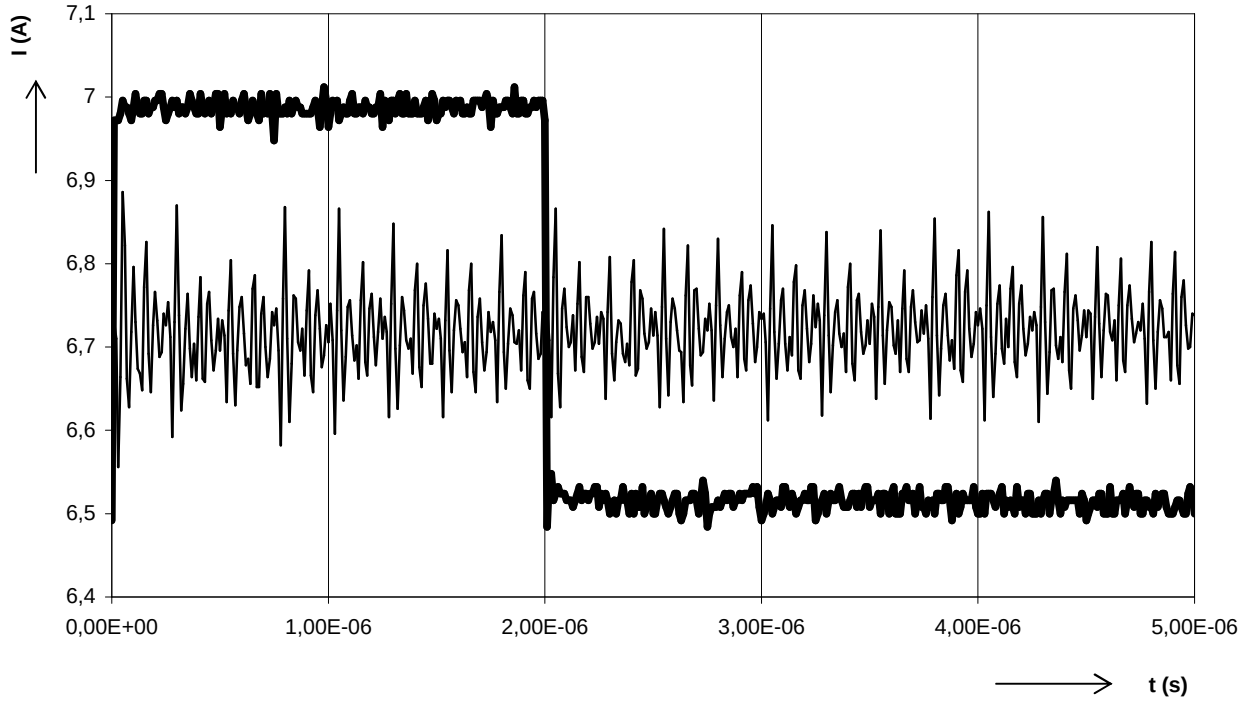
Měření probíhala na zapojení podle obrázku 5.1. Pro potlačení šumu a přesnější výsledky bylo každé měření opakováno desetkrát a následně zprůměrováno. Také byla pro eliminaci případného šumu na osciloskopu nastavena funkce průměrování. Do analyzovaného mikroprocesoru PIC16F84A byly postupně nahrány programy, viz kapitola 6.3. V jedné periodě proudového odběru mikroprocesoru, který zpracovává nahraný program, se provede pět instrukčních cyklů, což odpovídá dvaceti taktům oscilátoru. Grafy v kapitolách 6.4 až 6.7 jsou rovnoměrně rozděleny podle osy x na pět polí. Každé pole odpovídá jednomu instrukčnímu cyklu. Aby bylo možné jednotlivé instrukční cykly rozeznat a zjistit tak, kdy probíhá konkrétní instrukce, bylo nezbytné signál synchronizovat. Průběh signálu proudového odběru byl synchronizován se signálem z výstupního portu procesoru, díky tomu tak bylo možné přesně určit, v kterém intervalu probíhá určitá instrukce.

Na obrázku 6.4 je uvedeno porovnání taktů oscilátoru (tenká čára) s průběhem synchronizačního signálu (tlustá čára).

Obrázek 6.5 pak znázorňuje synchronizaci proudového odběru mikroprocesoru při vykonávání programu (tenká čára) se signálem z výstupního portu procesoru (tlustá čára) .




Obr. 6.4: Porovnání taktů oscilátoru s průběhem synchronizačního signálu.

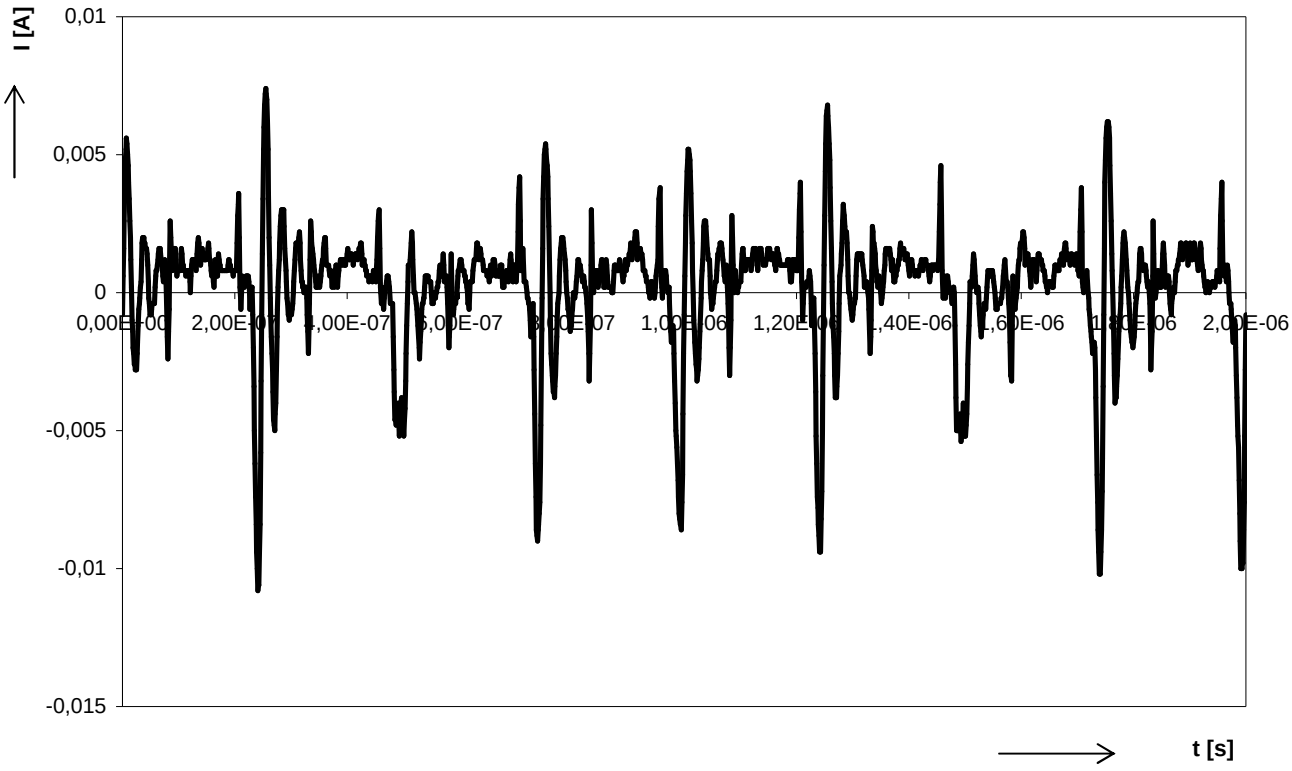



Obr. 6.5: Ukázka synchronizace proudového odběru se synchronizačním signálem.

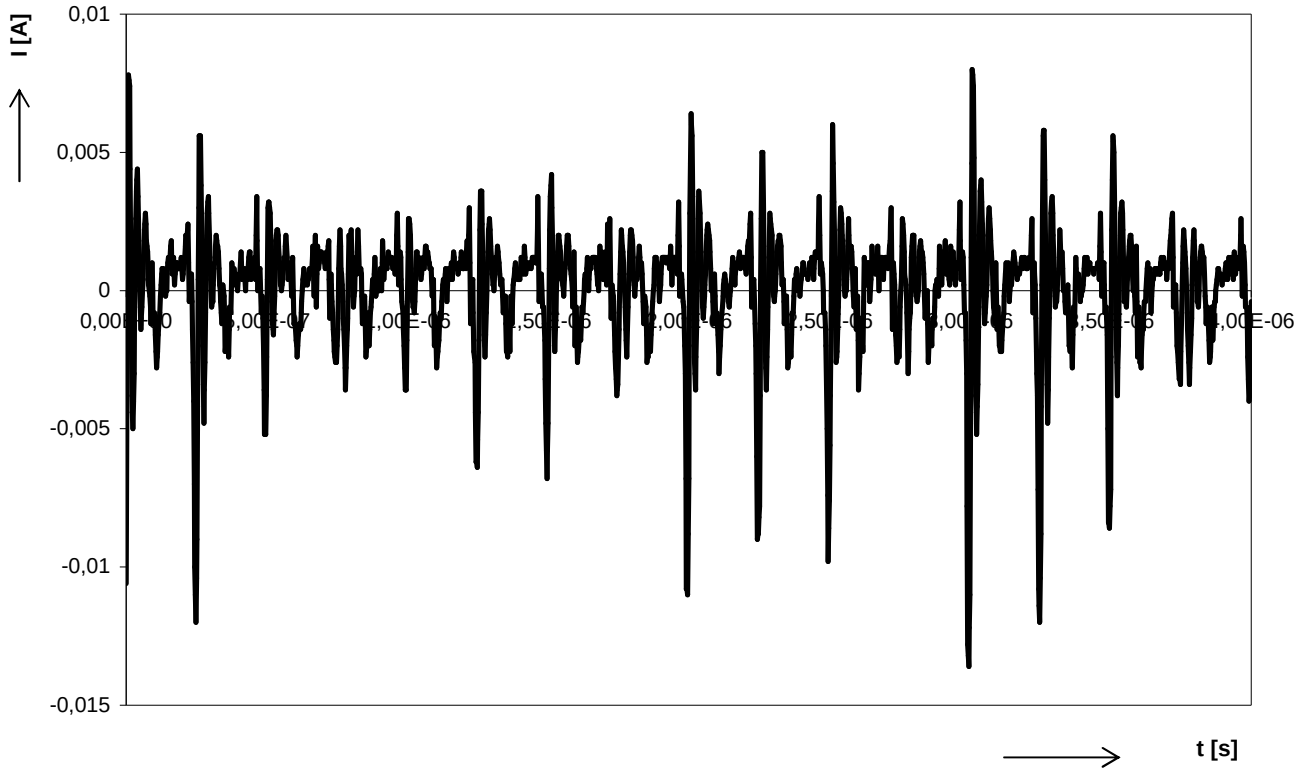
Na obrázku 6.6 je odběr proudu vymazaného a spuštěného mikroprocesoru PIC16F84A.

Z obrázku 6.7, kde je zobrazen průběh cyklicky se opakující instrukce XOR, je pak vidět, jak by to vypadalo, kdyby nebyla použita synchronizace.

Pokud by se signál synchronizoval se signálem z oscilátoru, viz obrázek 6.8, nebylo by možné instrukci přesně lokalizovat. První kanál osciloskopu tu představuje proudovou spotřebu instrukce XOR, za kterou následuje větvení programu. Druhý kanál osciloskopu pak zobrazuje signál oscilátoru. Z této ukázky je zřejmé, že signál z oscilátoru prakticky stále běží, což znemožňuje jakoukoliv orientaci v průběhu zobrazeném na prvním kanále osciloskopu. Proto, jak je uvedeno v kapitole 6.4, je v této práci pro synchronizaci použit signál z výstupního portu procesoru.




Obr. 6.6: Ukázka proudové spotřeby prázdného spuštěného mikroprocesoru.




Obr. 6.7: Ukázka proudová spotřeby zacyklené instrukce XOR bez synchronizace.


Obr. 6.8: Ukázka synchronizace proudového odběru se signálem s oscilátoru.

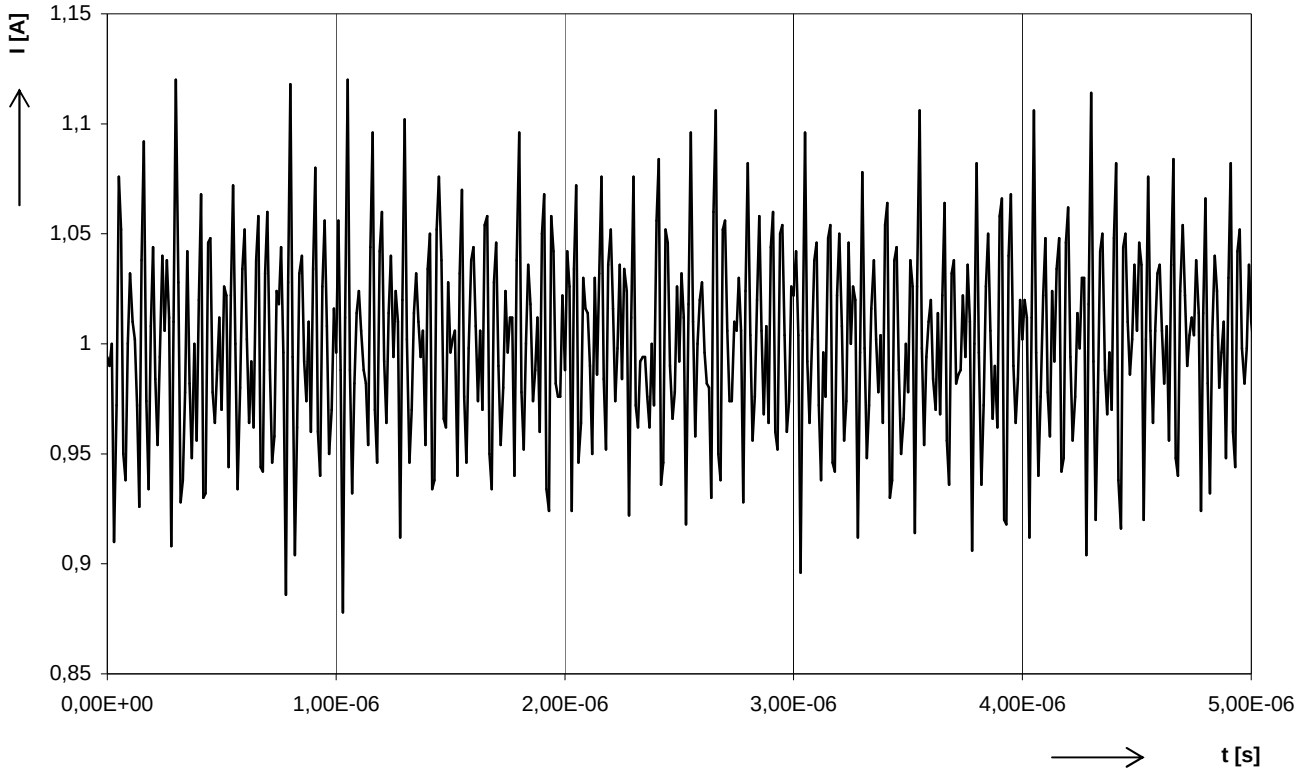
## 6.5 VÝBĚR MĚŘÍCÍ METODY

Protože všechny prostudované literatury, zabývající se výkonovou analýzou, obsahují pouze výsledná měření jiných algoritmů, které navíc zpracovávají jiné typy procesorů, a nezmiňují se o způsobu, postupu či návodu, kterým by bylo možné správně proudovou spotřebu procesoru při vykonávání jednotlivých instrukcí odměřit, bylo zapotřebí nejprve účinnou měřící metodu vybrat. Při analýze proudového postranního kanálu procesoru PIC16F84A bylo použito a odzkoušeno několik měřících metod, z nichž se nakonec vybrala ta nejúčinnější. Všechny vycházely ze zapojení podle obrázku 5.1.

### 6.5.1 PASIVNÍ SONDA

Nejprve byl osciloskop zapojen do obvodu pomocí pasivní sondy. Zemní vodič sondy však propojil obvod se zemí, což způsobilo zkratování přípravku s mikroprocesorem. Proto se vyzkoušelo sondu připojit mezi rezistor a vstup pro kladné napájecí napětí mikroprocesoru, ale výsledek nebyl uspokojivý, neboť bylo zřejmé, že zjištěné hodnoty napětí jsou příliš vysoké. Aby se dosáhlo korektní velikosti napětí na rezistoru, zkusilo se druhý kanál osciloskopu zapojit mezi rezistor a zdroj napájení. Pro odstranění případného šumu se na osciloskopu nastavila funkce průměrování. Poté se naměřené hodnoty na prvním a na druhém kanálu osciloskopu od sebe odečetly. Výsledný průběh byl ale značně ovlivněný šumem a hodnoty napětí byly po celé jeho délce téměř stejné. Šum se nepodařilo odstranit ani opakovanými měřeními, která se následně zprůměrovala. Tento postup tedy není vhodný pro měření malých napětí, což plyne i z [5].

Ukázka proudové spotřeby procesoru při zpracování programu XOR, která byla odměřena touto metodou je na obrázku 6.9.

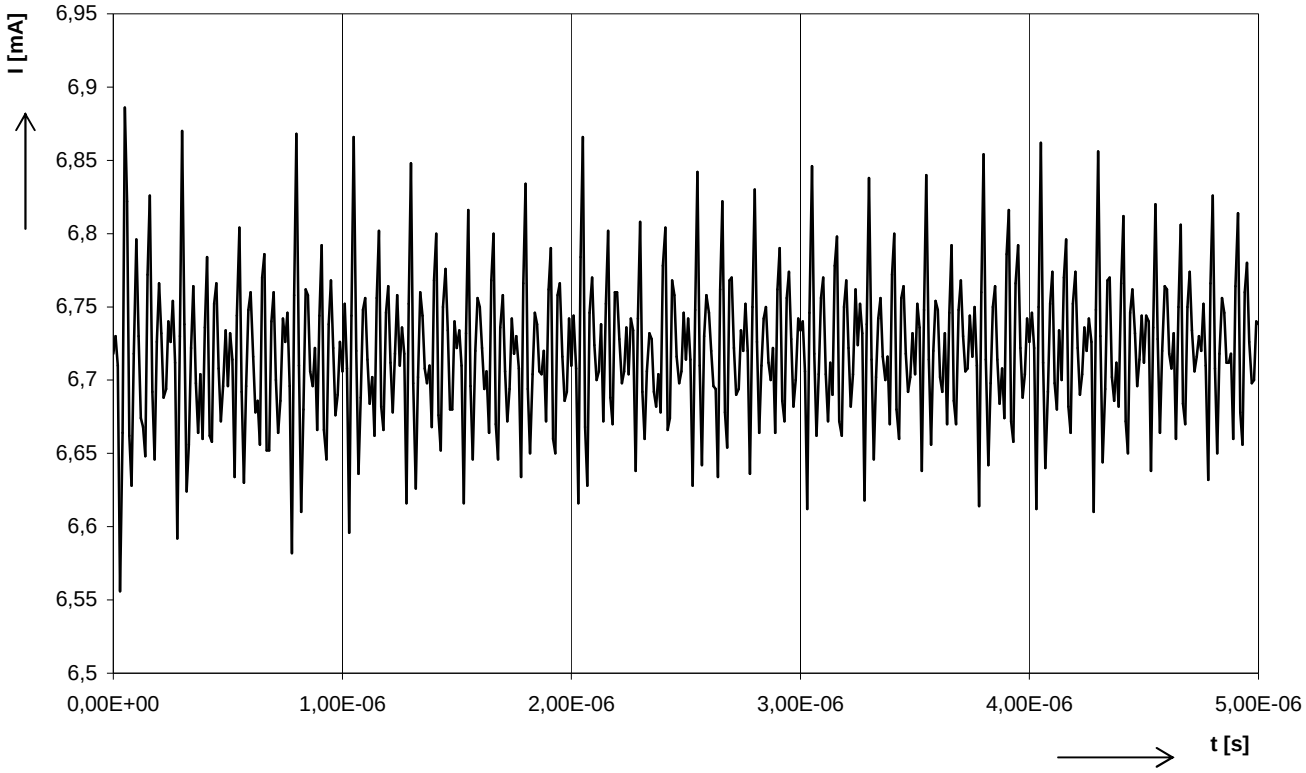



Obr. 6.9: Proudový odběr programu XOR měřený dvěma pasivními sondami.

### 6.5.2 DIFERENČNÍ SONDA

V další fázi výzkumu se pasivní sonda nahradila sondou diferenční. Tu již bylo možné zapojit podle obrázku 5.1, tedy přímo na rezistor. Po připojení do obvodu již sice zkrat způsoben nebyl, ale naměřené hodnoty napětí byly velmi nízké. Velikost tohoto útlumu byla dána vlastnostmi použité diferenční sondy. Navíc byl výsledný signál, stejně jako u předchozí metody, příliš ovlivněný šumem. Utlumený průběh bylo sice možné z nastaveného útlumu na sondě přepočítat tak, aby hodnoty odpovídali jejich skutečné velikosti. Tím se ale zároveň zvedla i velikost šumu a průběh byl nadále po celé délce téměř stejný, viz obr. 6.10. Opět k odstranění šumu nepomohlo ani několikanásobné opakování měření s následným zprůměrováním. I u této metody byla, pro odstranění eventuálního šumu, na osciloskopu nastavena funkce průměrování.

Jelikož nebyla k dispozici jiná diferenční sonda, která by měřila s menším nebo ideálně s žádným útlumem, nebylo možné učinit další pokusy a metoda byla opět označena za nevhodnou.



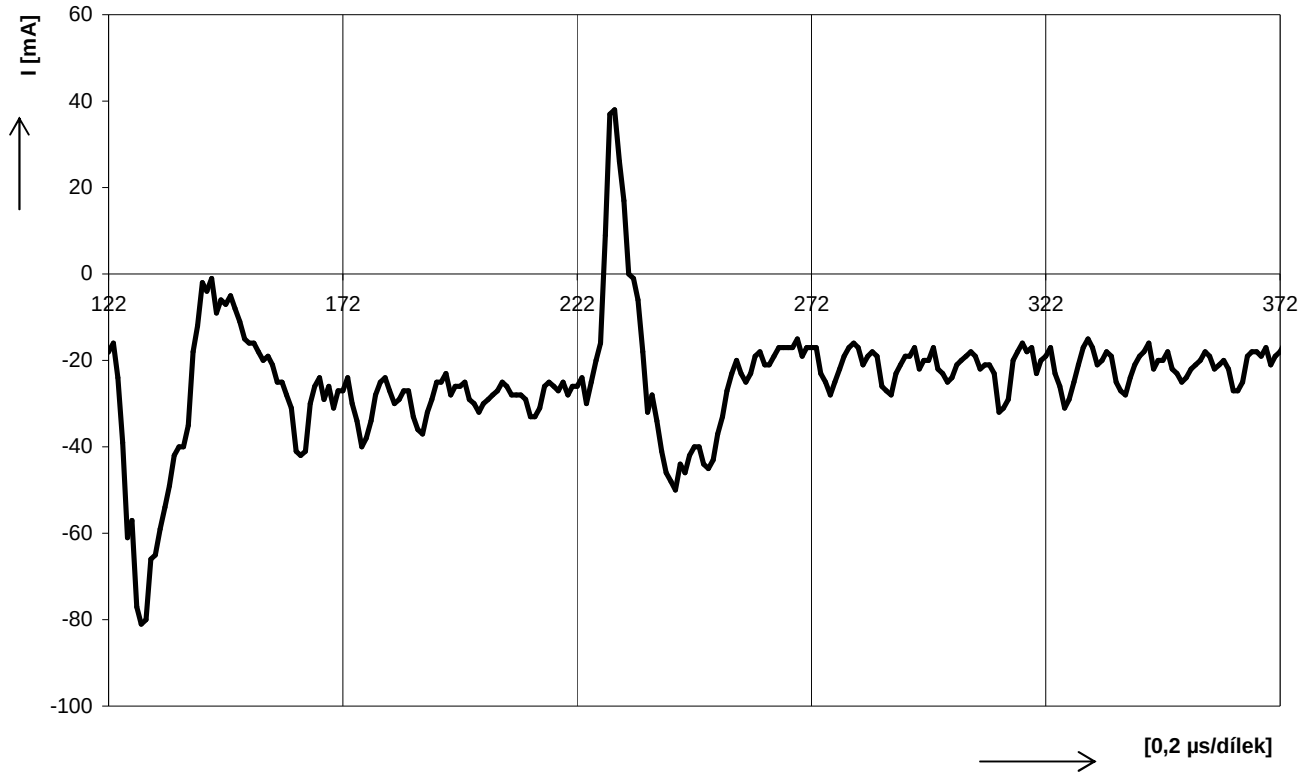

Obr. 6.10: Proudový odběr programu XOR měřený diferenční sondou.

### 6.5.3 BATERIOVÝ OSCILOSKOP

Dalšími pokusy a zkoušením se došlo ke způsobu, který umožňuje obvod zapojit podle obrázku 5.1, aniž by ho zkratoval, a zároveň výsledné hodnoty příliš nezkresluje. Toho je možné docílit použitím digitálního bateriového osciloskopu, který je napájen z baterie a má tedy oddělenou zem od napájení. Před zapojením sondy do obvodu je ale důležité, aby se vždy po nahrání programu do procesoru odpojil datový kabel od desky PICDEM$^{TM}$ 2 PLUS, jinak by byla z počítačem propojená přes port USB, který by ji spojoval se zemí a opět by tak došlo ke zkratování obvodu. Po odpojení datového kabelu od přípravku se program nahraný v mikroprocesoru automaticky spustí.

Ukázka měření proudového odběru mikroprocesoru při zpracování programu XOR, která byla odměřena touto metodou je na obrázku 6.11. Při tomto měření byl pro lepši zobrazení použit oscilátor s taktem 20 MHz.

Tyto hodnoty již sice jsou pro analýzu použitelné, ale protože byl tento typ osciloskopu zapůjčený pouze na krátkou dobu a jiný nebyl k dispozici, nebyla tato metoda pro experimentální část nakonec využita.




Obr. 6.11: Proudový odběr programu XOR měřený bateriovým osciloskopem.

### 6.5.4 ODDĚLOVACÍ TRANSFORMÁTOR

Tato metoda souvisí s použitím oddělovacího transformátoru, do kterého se připojí digitální osciloskop. I zde je odděleno uzemnění od napájení a po připojení sondy do obvodu nedochází k jeho zkratování. Postup pro zapojení sondy do obvodu je stejný jako u bateriového osciloskopu, opět se tedy musí nejprve odpojit datový kabel. Protože zde nedochází k téměř žádnému nežádoucímu ovlivnění signálu šumem, je tato měřící metoda při zpracování této práce nakonec využita.

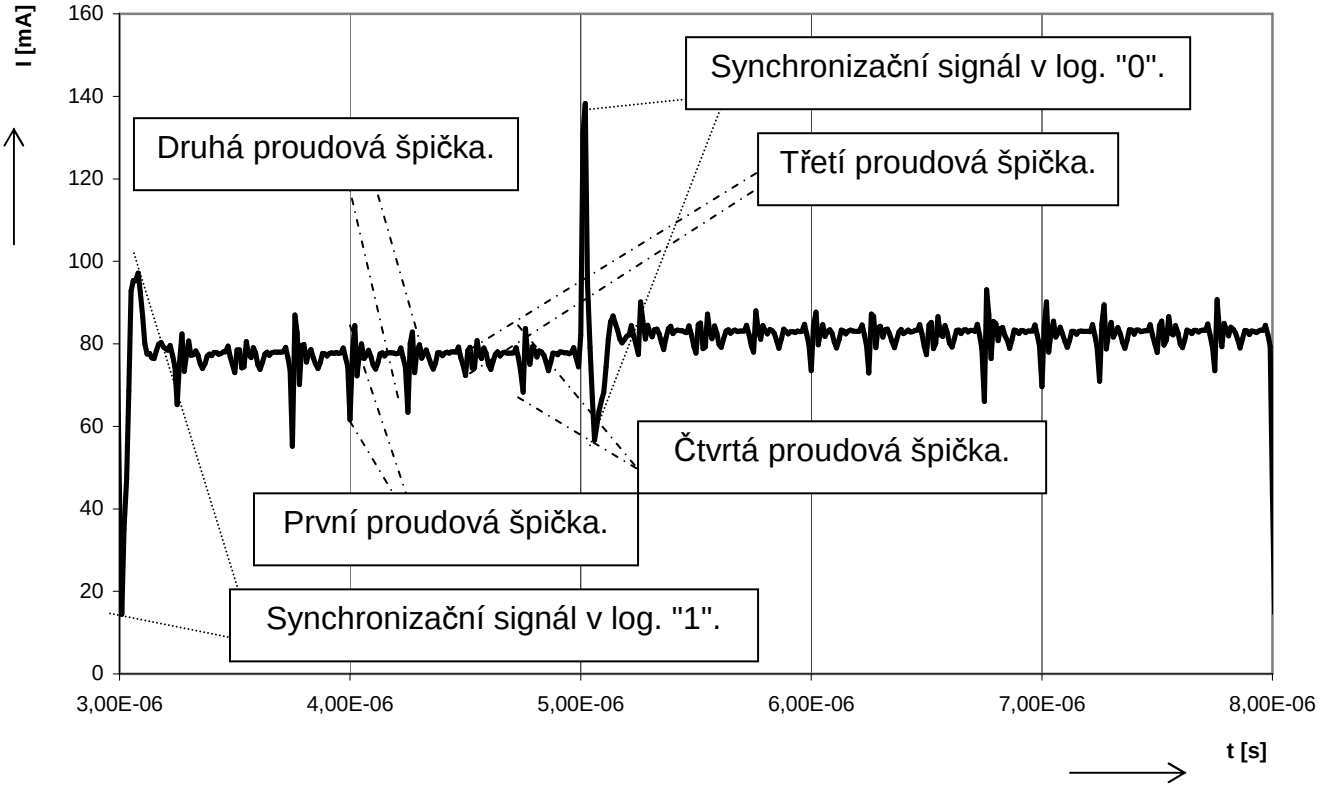
Z grafů v následujících kapitolách je zřejmé, že výsledné hodnoty naměřené tímto způsobem jsou srovnatelné s hodnotami získanými pomocí bateriového osciloskopu. Obě tyto metody lze tedy pro měření proudového odběru mikroprocesoru použít.

## 6.6 VLASTNÍ MĚŘENÍ

Měření proudového odběru mikroprocesoru PIC16F84A při vykonávání jednotlivých programů popsaných v kapitole 6.3, se nejprve provedlo pro oscilátor s taktem 4 MHz. Protože ale velikost proudové spotřeby závisí jak na napájecím napětí, tak i na frekvenci použitého oscilátoru, zopakovalo se měření také pro oscilátor s taktem 20 MHz. Byla použita měřící metoda využívající oddělovací transformátor, viz kapitola 6.5.4. Synchronizační signál z výstupního portu procesoru, viz kapitola 6.4, není pro lepší zobrazení spotřeby proudu zobrazen. Pro pohyb v kladných číslech, jsou všechny hodnoty posunuty o stejnosměrnou složku 80 mV, která však výsledky měření nikterak neovlivňuje.

### 6.6.1 OSCILÁTOR 4 MHz

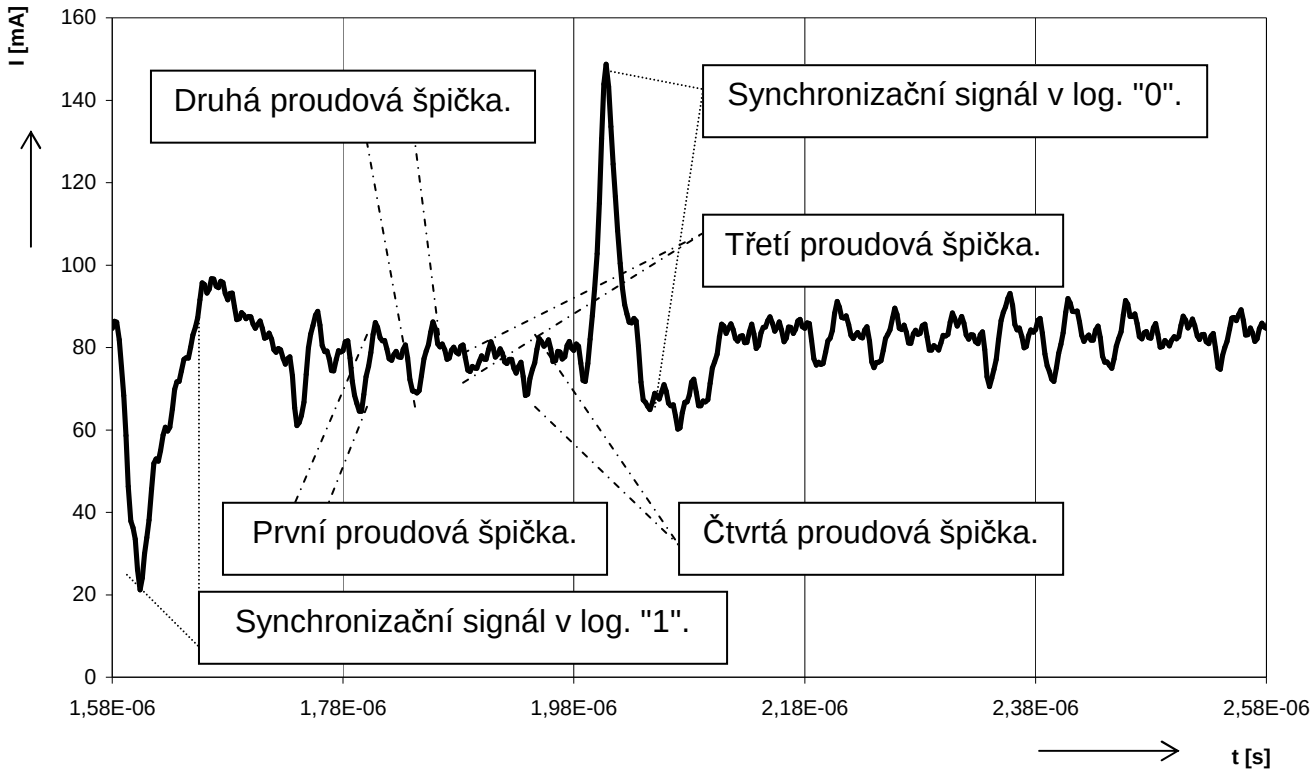
Tato podkapitola obsahuje výstup měření proudové spotřeby procesoru PIC16F84A při vykonávání programu XOR (Obr. 6.12). Do obvodu je zapojený oscilátor s frekvencí 4 MHz. V grafu jsou znázorněny nejdůležitější proudové špičky využité v kapitole 6.7 pro analýzu výsledků.




Obr. 6.12: Proudová spotřeba programu XOR.

### 6.6.2 OSCILÁTOR 20 MHz

V této podkapitole je znázorněn výstup měření proudové spotřeby procesoru PIC16F84A při vykonávání programu XOR (Obr. 6.13). Do obvodu je zapojený oscilátor s frekvencí 20 MHz. V grafu jsou znázorněny nejdůležitější proudové špičky využité v kapitole 6.7 pro analýzu výsledků.




Obr. 6.13: Proudová spotřeba programu XOR.

Výsledné grafy pro oscilátor s frekvencí 4 MHz i 20 MHz jsou pro všechny programy podobné, proto je u obou případů zobrazen pouze jeden vzorový graf, ostatní grafy je možné prohlédnout v elektronické příloze.

## 6.7 ANALÝZA VÝSLEDKŮ

Jak je uvedeno v kapitole 6.4, všechny grafy jsou rovnoměrně rozděleny podle osy x na pět polí, z nichž každé pole odpovídá jednomu instrukčnímu cyklu a ten odpovídá konkrétní instrukci. Díky tomu je možné přesně určit, co v daném intervalu právě probíhá a porovnat mezi sebou odpovídající si hodnoty proudu.

Pro oscilátor s taktem 4 MHz platí, že každé pole trvá 1 µs, což tedy znamená, že se program provede za 5 µs.

Pro oscilátor s taktem 20 MHz platí, že jedno pole trvá 0,2 µs, což v tomto případě znamená, že se program provede za 1 µs.

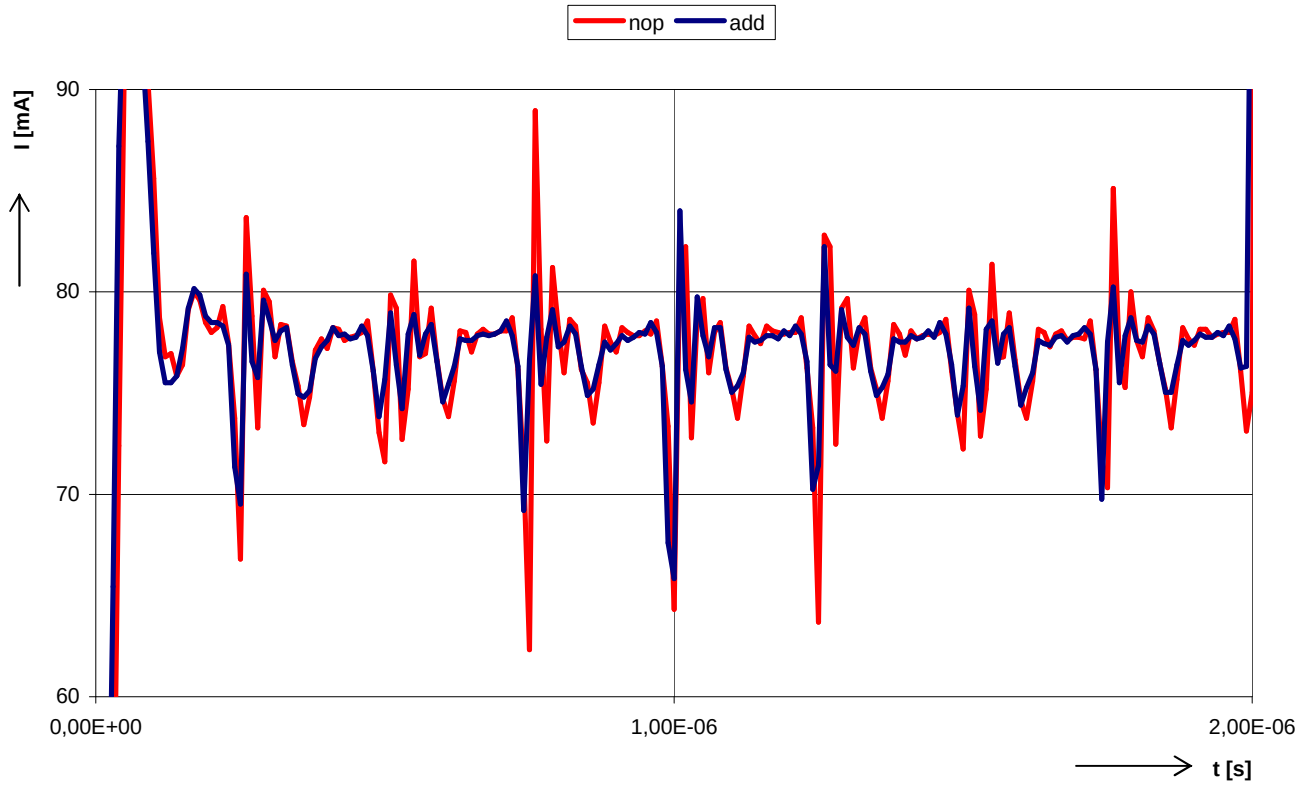
V prvním poli se synchronizační signál nastaví do stavu logická "1", ve druhém poli se podle typu programu provede daná instrukce, ve třetím poli se synchronizační signál nastaví do stavu logická "0", ve čtvrtém a pátém poli se provede větvení programu a skok na začátek, což odpovídá dvěma instrukčním cyklům, přičemž v pátém poli se provede prázdná instrukce NOP.

Při slovní analýze je pozornost věnována pouze prvním třem polím. Při porovnání ve grafech jsou pak vykresleny jen první dvě pole, přičemž první pole je zobrazeno jen pro případ, že by instrukci začal procesor zpracovávat už v něm. V kapitolách 6.7.1 a 6.7.2 není v grafech zobrazena proudová špička synchronizačního signálu, protože je rozsah osy y úmyslně volený tak, aby byly rozdíly v průbězích proudové spotřeby patřící jednotlivým instrukcím zřetelnější.

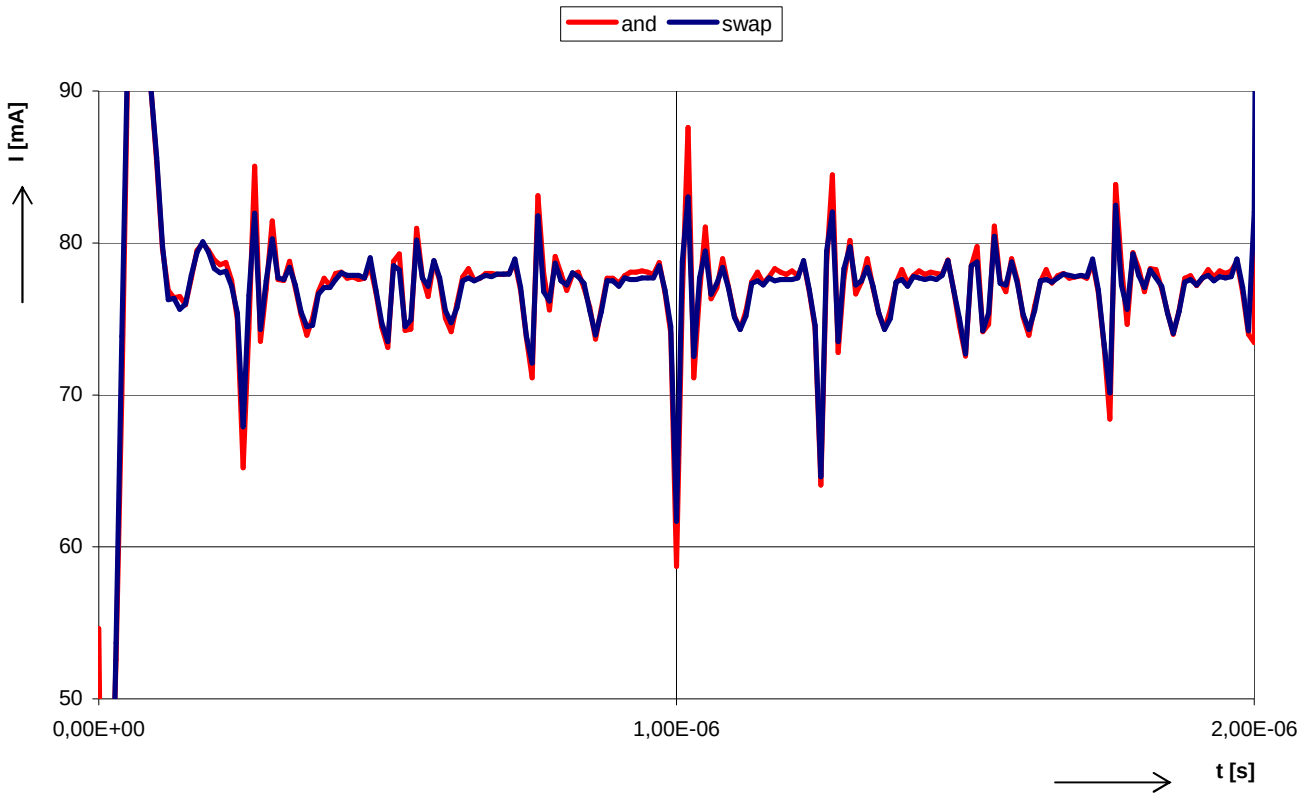
### 6.7.1 PRO OSCILÁTOR 4 MHZ

Na obrázku 6.14 je uveden průběh proudové spotřeby všech šesti instrukcí. Ale protože se jednotlivé průběhy překrývají, jsou kvůli lepší přehlednosti náhodně rozděleny do tří skupin po dvou. Na obrázku 6.15 jsou zobrazeny průběhy instrukcí NOP a ADD, na obrázku 6.16. jsou průběhy instrukcí AND a SWAP, a na obrázku 6.17 jsou průběhy instrukcí XOR a INC.

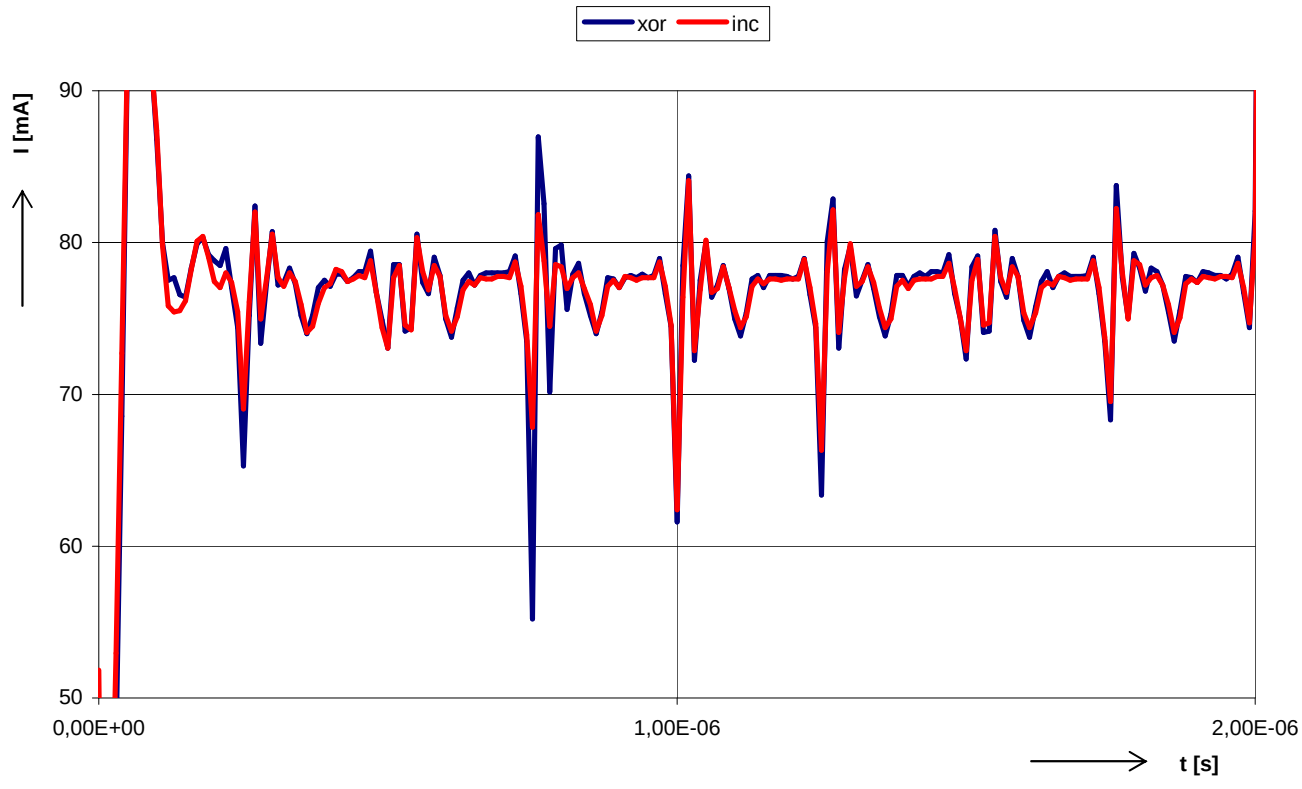

Obr. 6.14: Proudová spotřeba všech instrukcí.




Obr. 6.15: Proudová spotřeba instrukcí NOP a ADD.




Obr. 6.16: Proudová spotřeba instrukcí AND a SWAP.




Obr. 6.17: Proudová spotřeba instrukcí XOR a INC.

- **Program ADD.**

Proudová špička synchronizačního signálu ve stavu logická "1" je 73,76 mA. Proudová špička synchronizačního signálu ve stavu logická "0" je 86,24 mA. Proudová špička mezi těmito stavy je 119,44 mA.

Ve druhém poli se provádí instrukce ADD, která provádí operaci sčítání. První proudová špička má velikost 18,16 mA, druhá 12,00 mA, třetí 3,84 mA a čtvrtá 10,48 mA. Instrukce se provádí přibližně při proudovém odběru 78,00 mA.

- **Program XOR.**

Proudová špička synchronizačního signálu ve stavu logická "1" je 82,72 mA. Proudová špička synchronizačního signálu ve stavu logická "0" je 81,60 mA. Proudová špička mezi těmito stavy je 123,84 mA.

Ve druhém poli se provádí instrukce XOR, která provede exkluzivní logický součet. První proudová špička má velikost 22,80 mA, druhá 19,52 mA, třetí 6,80 mA a čtvrtá 15,44 mA. Instrukce se provádí přibližně při proudovém odběru 78,00 mA.

- **Program INC.**

Proudová špička synchronizačního signálu ve stavu logická "1" je 78,80 mA. Proudová špička synchronizačního signálu ve stavu logická "0" je 77,68 mA. Proudová špička mezi těmito stavy je 113,34 mA.

Ve druhém poli se provádí instrukce INC, která zvětší obsah registru o jedničku. První proudová špička má velikost 21,68 mA, druhá 15,84 mA, třetí 6,00 mA a čtvrtá 12,72 mA. Instrukce se provádí přibližně při proudovém odběru 78,00 mA.

- **Program NOP.**

Proudová špička synchronizačního signálu ve stavu logická "1" je 75,12 mA. Proudová špička synchronizačního signálu ve stavu logická "0" je 77,92 mA. Proudová špička mezi těmito stavy je 114,64 mA.

Ve druhém poli se provádí instrukce NOP, neudělá se tedy nic. První proudová špička má velikost 17,92 mA, druhá 19,12 mA, třetí 7,64 mA a čtvrtá 14,80 mA. Instrukce se provádí přibližně při proudovém odběru 78,00 mA.

- **Program SWAP.**

Proudová špička synchronizačního signálu ve stavu logická "1" je 81,90 mA. Proudová špička synchronizačního signálu ve stavu logická "0" je 84,20 mA. Proudová špička mezi těmito stavy je 126,10 mA.

Ve druhém poli se provádí instrukce SWAP, prohodí horní a dolní půlbyte regitru. První proudová špička má velikost 21,30 mA, druhá 17,40 mA, třetí 6,10 mA a čtvrtá 12,40 mA. Instrukce se provádí přibližně při proudovém odběru 78,00 mA.

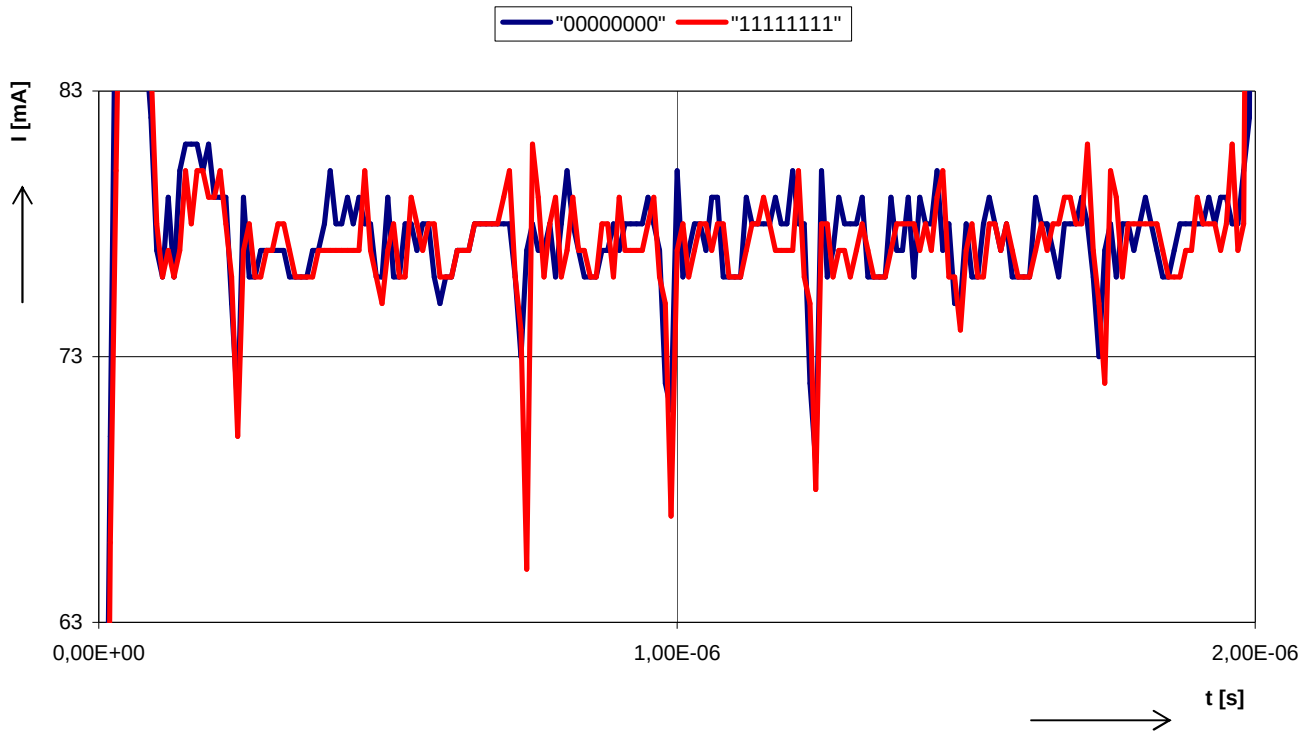
- **Program AND.**

Proudová špička synchronizačního signálu ve stavu logická "1" je 74,68 mA. Proudová špička synchronizačního signálu ve stavu logická "0" je 75,68 mA. Proudová špička mezi těmito stavy je 113,60 mA.

Ve druhém poli se provádí instrukce AND, která vykoná logický součin. První proudová špička má velikost 28,88 mA, druhá 20,40 mA, třetí 7,20 mA a čtvrtá 15,44 mA. Instrukce se provádí přibližně při proudovém odběru 78,00 mA.

- **Porovnání instrukce XOR s různou binární hodnotou vstupních dat.**

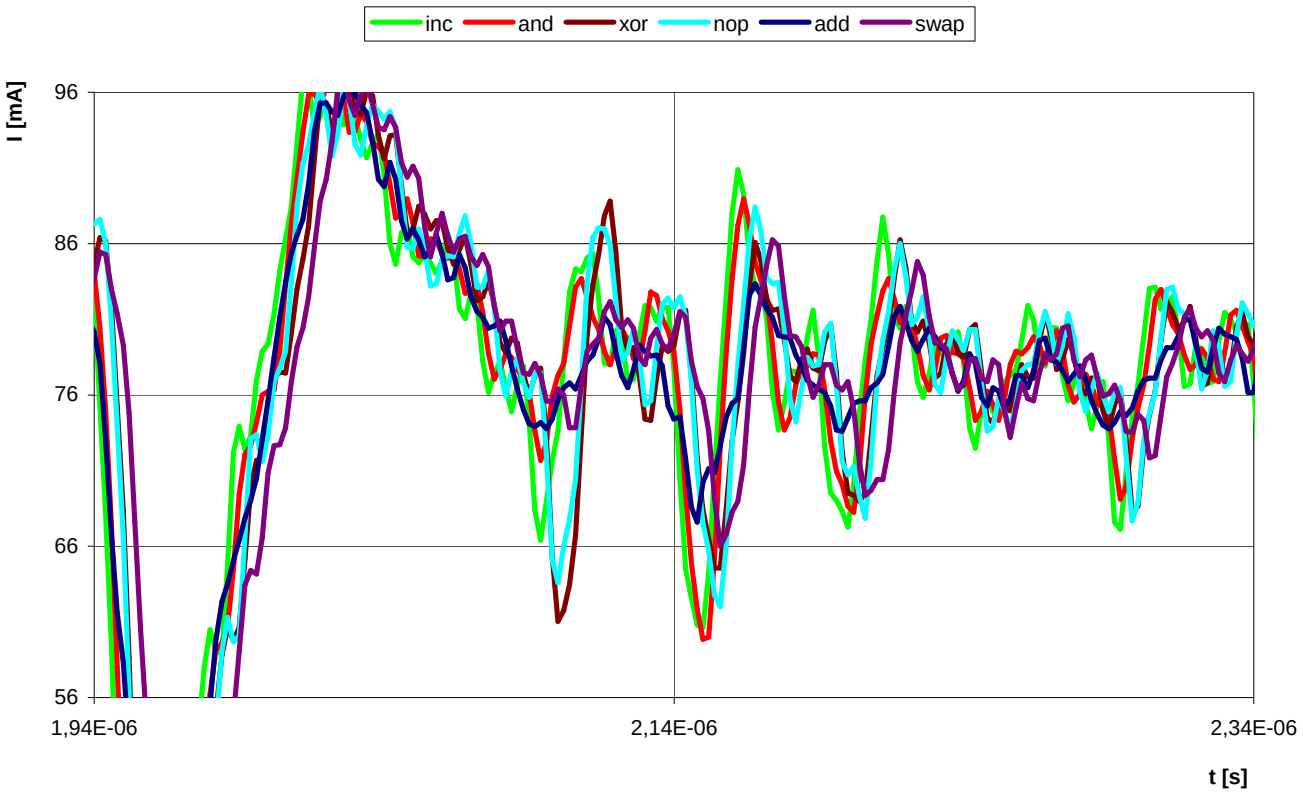
Protože se proudová spotřeba instrukce AND s odlišnou binární hodnotou vstupních dat od ostatních instrukcí výrazně nelišila, provedla se další měření programu XOR s různými vstupními daty, aby se ověřil jejich vliv na výsledný průběh. Operace exkluzivního logického součtu se postupně prováděla s různým počtem jedniček v binární kombinaci, a to v rozmezí od 00000000 do 11111111. Na obrázku 6.18 je porovnání odběru proudu právě těchto kombinací. Průběhy ostatních kombinací jsou podobné a lze si je prohlédnout v elektronické příloze.



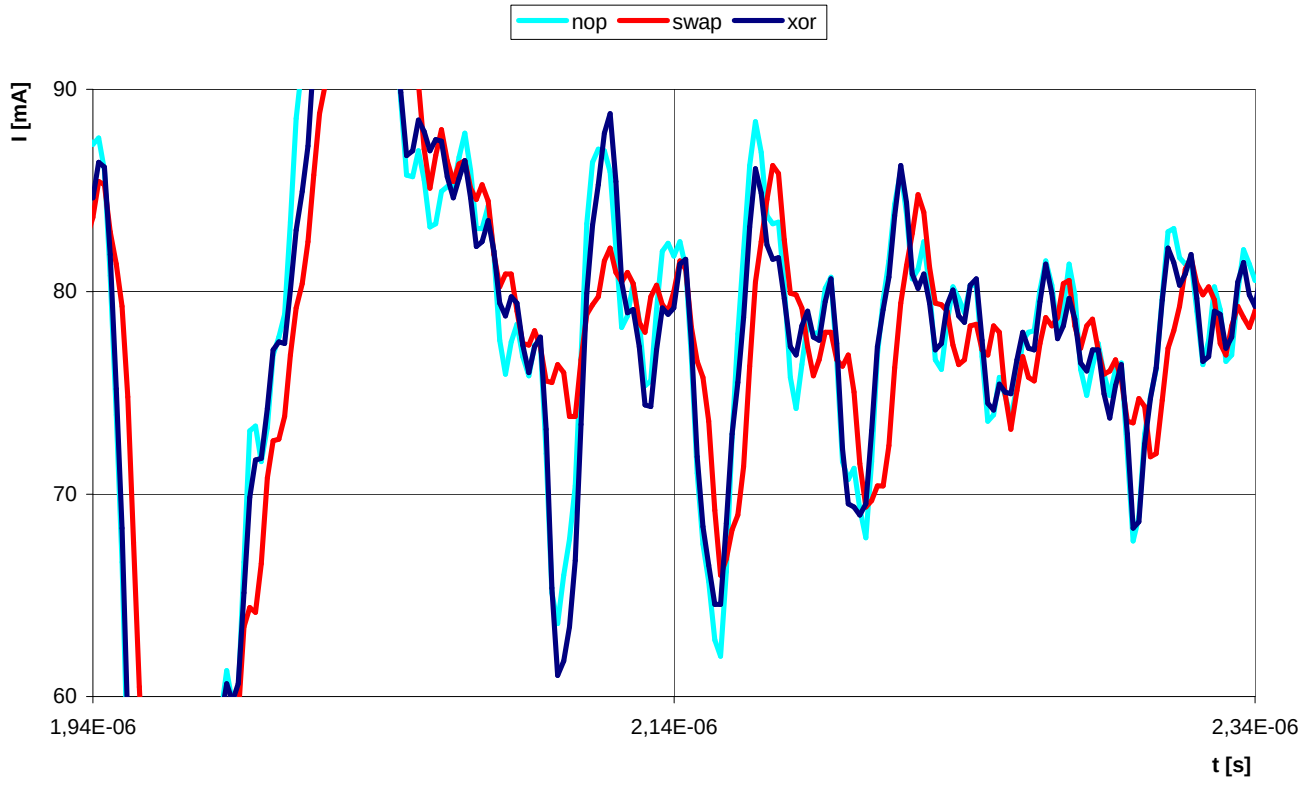

Obr. 6.18: Porovnání proudové spotřeby instrukce XOR s různou vstupní binární hodnotou.

### 6.7.2 PRO OSCILÁTOR 20 MHz

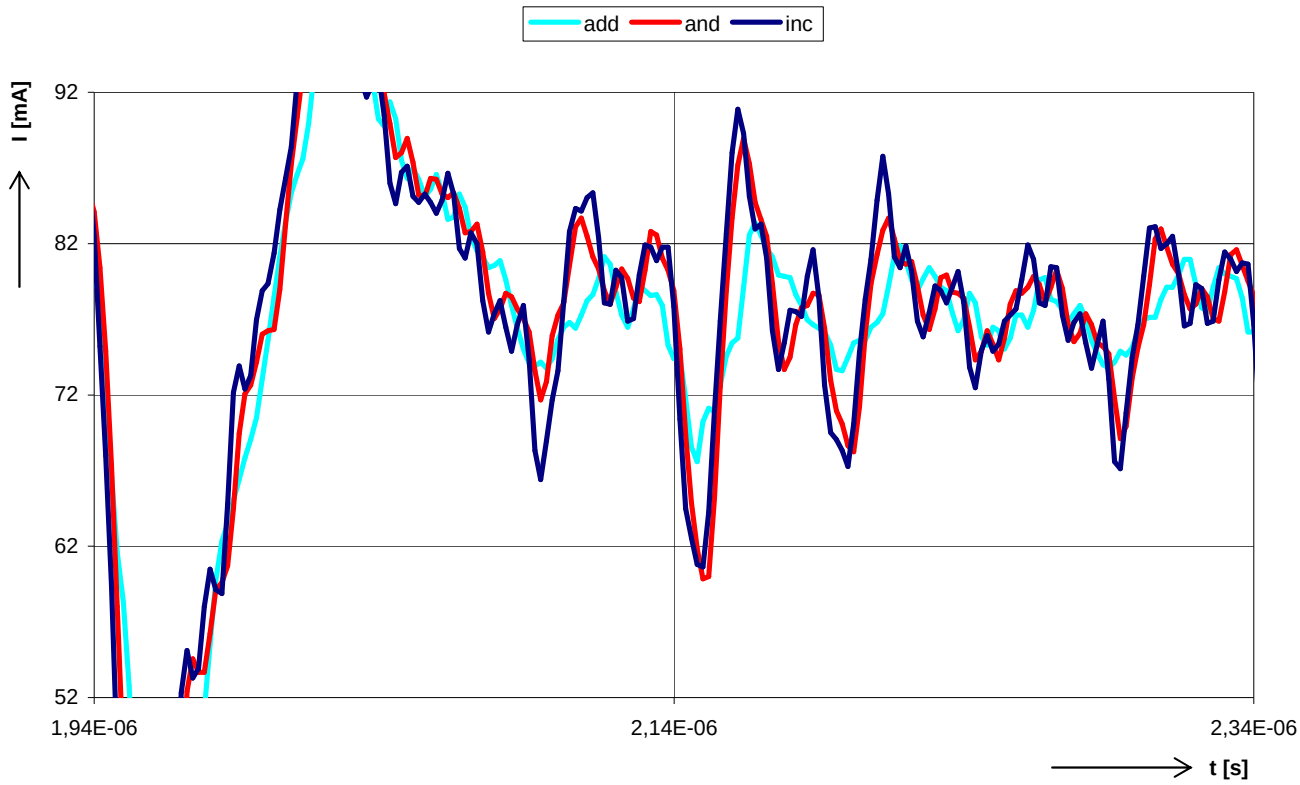
Na obrázku 6.19 je uveden průběh proudové spotřeby všech šesti instrukcí. Ale protože všechny průběhy v jednom grafu nejsou příliš přehledné, i když je to znatelně lepší než v případě oscilátoru s taktem 4 MHz, jsou v tomto případě pro lepší přehlednost náhodně rozděleny do dvou skupin po třech. Na obrázku 6.20 jsou průběhy instrukcí NOP, SWAP a XOR, na obrázku 6.21 jsou pak zobrazeny průběhy instrukcí ADD, AND a INC.




Obr. 6.19: Proudová spotřeba všech instrukcí.




Obr. 6.20: Proudová spotřeba instrukcí NOP, SWAP a XOR.




Obr. 6.21: Proudová spotřeba instrukcí ADD, AND a INC.

- **Program ADD.**

Proudová špička synchronizačního signálu ve stavu logická "1" je 59,30 mA. Proudová špička synchronizačního signálu ve stavu logická "0" je 59,60 mA. Proudová špička mezi těmito stavy je 88,60 mA.

Ve druhém poli se provádí instrukce ADD, která provádí operaci sčítání. První proudová špička má velikost 15,80 mA, druhá 8,20 mA, třetí 4,00 mA a čtvrtá 6,80 mA. Instrukce se provádí přibližně při proudovém odběru 80,00 mA.

- **Program INC.**

Proudová špička synchronizačního signálu ve stavu logická "1" je 78,60 mA. Proudová špička synchronizačního signálu ve stavu logická "0" je 93,40 mA. Proudová špička mezi těmito stavy je 132,60 mA.

Ve druhém poli se provádí instrukce INC, která zvětší obsah registru o jedničku. První proudová špička má velikost 30,30 mA, druhá 20,50 mA, třetí 8,50 mA a čtvrtá 16,00 mA. Instrukce se provádí přibližně při proudovém odběru 80,00 mA.

- **Program NOP.**

Proudová špička synchronizačního signálu ve stavu logická "1" je 78,10 mA. Proudová špička synchronizačního signálu ve stavu logická "0" je 87,90 mA. Proudová špička mezi těmito stavy je 128,70 mA.

Ve druhém poli se provádí instrukce NOP, neudělá se tedy nic. První proudová špička má velikost 26,40 mA, druhá 18,20 mA, třetí 7,60 mA a čtvrtá 15,40 mA. Instrukce se provádí přibližně při proudovém odběru 80,00 mA.

- **Program SWAP.**

Proudová špička synchronizačního signálu ve stavu logická "1" je 74,10 mA. Proudová špička synchronizačního signálu ve stavu logická "0" je 84,00 mA. Proudová špička mezi těmito stavy je 127,80 mA.

Ve druhém poli se provádí instrukce SWAP, prohodí horní a dolní půlbyte regitru. První proudová špička má velikost 21,50 mA, druhá 16,70 mA, třetí 7,40 mA a čtvrtá 12,80 mA. Instrukce se provádí přibližně při proudovém odběru 80,00 mA.

- **Program XOR.**

Proudová špička synchronizačního signálu ve stavu logická "1" je 74,50 mA. Proudová špička synchronizačního signálu ve stavu logická "0" je 86,00 mA. Proudová špička mezi těmito stavy je 127,80 mA.

Ve druhém poli se provádí instrukce XOR, která provede exkluzivní logický součet. První proudová špička má velikost 21,50 mA, druhá 16,70 mA, třetí 7,20 mA a čtvrtá 13,60 mA. Instrukce se provádí přibližně při proudovém odběru 80,00 mA.

- **Program AND.**

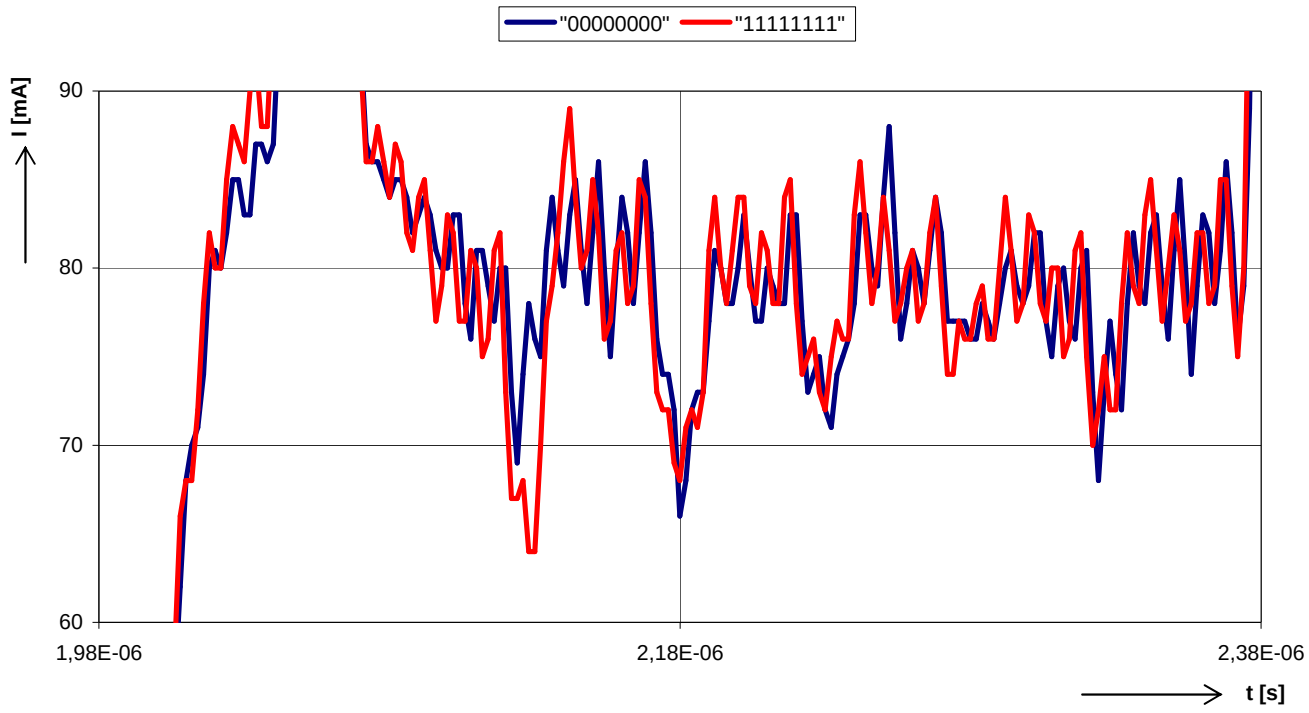
Proudová špička synchronizačního signálu ve stavu logická "1" je 78,70 mA. Proudová špička synchronizačního signálu ve stavu logická "0" je 87,00 mA. Proudová špička mezi těmito stavy je 125,60 mA.

Ve druhém poli se provádí instrukce AND, která vykoná logický součin. První proudová špička má velikost 29,00 mA, druhá 15,50 mA, třetí 4,40 mA a čtvrtá 13,90 mA. Instrukce se provádí přibližně při proudovém odběru 80,00 mA.

- **Porovnání instrukce XOR s různou binární hodnotou vstupních dat.**

Stejné měření jako pro oscilátor 4 MHz se provedlo i pro oscilátor 20 MHz. Na obrázku 6.22 je porovnání odběru proudu instrukce XOR, která se provádí nejprve s binární kombinaci 00000000 a následně s binární kombinaci 11111111.

Průběhy ostatních kombinací jsou znovu podobné a lze si je prohlédnout v elektronické příloze.




Obr. 6.22: Porovnání proudové spotřeby instrukce XOR s různou vstupní binární hodnotou.

### 6.7.3 4 MHz vs. 20 MHz

Hodnoty jednotlivých proudových špiček z podkapitoly 6.7.1 pro první tři bloky jsou kvůli přehlednosti sepsány do tabulky 6.1 a následně setříděny sestupně podle velikosti odebíraného proudu v jednotlivých fázích průběhu.

Stejně tak tabulka 6.2 obsahuje hodnoty proudu z podkapitoly 6.7.2, které jsou stejně jako v předešlém případě setříděny sestupně.

Z tabulek je tedy zřetelně vidět, která instrukce má v konkrétní fázi průběhu nejvyšší proudovou spotřebu a která ji má naopak nejnižší.

Tab. 6.1: Porovnání proudového odběru jednotlivých programů pro oscilátor 4 MHz.

| Program | Synchronizační signál ve stavu logická "1" [mV] | Synchronizační signál ve stavu logická "0" [mV] | Mezi těmito stavy [mV] | 1. proudová špička [mV] | 2. proudová špička [mV] | 3. proudová špička [mV] | 4. proudová špička [mV] |
|---|---|---|---|---|---|---|---|
| **ADD** | 73,76 | 86,24 | 119,44 | 18,16 | 12,00 | 3,84 | 10,48 |
| **XOR** | 82,72 | 81,60 | 123,84 | 22,80 | 19,52 | 6,80 | 15,44 |
| **INC** | 78,80 | 77,68 | 113,34 | 21,68 | 15,84 | 6,00 | 12,72 |
| **NOP** | 75,12 | 77,92 | 114,64 | 17,92 | 19,12 | 7,64 | 14,80 |
| **SWAP** | 81,90 | 84,20 | 126,10 | 21,30 | 17,40 | 6,10 | 12,40 |
| **AND** | 74,68 | 75,68 | 113,60 | 28,88 | 20,40 | 7,20 | 15,44 |
| **1. největší odběr [mV]** | XOR | ADD | SWAP | AND | AND | NOP | AND |
| **2. největší odběr [mV]** | SWAP | SWAP | XOR | XOR | XOR | AND | XOR |
| **3. největší odběr [mV]** | INC | XOR | ADD | INC | NOP | XOR | NOP |
| **4. největší odběr [mV]** | NOP | NOP | NOP | SWAP | SWAP | SWAP | SWAP |
| **5. největší odběr [mV]** | AND | INC | AND | ADD | INC | INC | INC |
| **6. největší odběr [mV]** | ADD | AND | INC | NOP | ADD | ADD | ADD |

Z tabulky 6.1 vyplívá, že pro procesor PIC16F84A je nejnáročnější instrukce AND. Pomyslné druhé místo by obsadila instrukce XOR. Naopak určení nejméně náročné instrukce už tak jednoduché není. K teorii se nejvíce blíží setřídění instrukcí ve sloupci 1. proudová špička, což by znamenalo, že nejméně náročnou instrukcí je instrukce NOP. Pokud by se vycházelo ze všech čtyř proudových špiček, byla by to instrukce ADD.

Tab. 6.2: Porovnání proudového odběru jednotlivých programů pro oscilátor 20 MHz.

| **Program** | **Synchronizační signál ve stavu logická "1" [mV]** | **Synchronizační signál ve stavu logická "0" [mV]** | **Mezi těmito stavy [mV]** | **1. proudová špička [mV]** | **2. proudová špička [mV]** | **3. proudová špička [mV]** | **4. proudová špička [mV]** |
|---|---|---|---|---|---|---|---|
| **ADD** | 59,30 | 59,60 | 88,60 | 15,80 | 8,20 | 4,00 | 6,80 |
| **INC** | 78,60 | 93,40 | 132,60 | 30,30 | 20,50 | 8,50 | 16,00 |
| **NOP** | 78,10 | 87,90 | 128,70 | 26,40 | 18,20 | 7,60 | 15,40 |
| **SWAP** | 74,10 | 84,00 | 127,80 | 21,50 | 16,70 | 7,40 | 12,80 |
| **XOR** | 74,50 | 86,00 | 127,80 | 21,50 | 16,70 | 7,20 | 13,60 |
| **AND** | 78,70 | 87,00 | 125,60 | 29,00 | 15,50 | 4,40 | 13,90 |
| **1. největší odběr [mV]** | AND | INC | INC | INC | INC | INC | INC |
| **2. největší odběr [mV]** | INC | NOP | NOP | AND | NOP | NOP | NOP |
| **3. největší odběr [mV]** | NOP | AND | XOR | NOP | XOR | SWAP | AND |
| **4. největší odběr [mV]** | XOR | XOR | SWAP | XOR | SWAP | XOR | XOR |
| **5. největší odběr [mV]** | SWAP | SWAP | AND | SWAP | AND | AND | SWAP |
| **6. největší odběr [mV]** | ADD | ADD | ADD | ADD | ADD | ADD | ADD |

Podle tabulky 6.2 je nejnáročnější instrukcí instrukce INC, druhou nejnáročnější instrukcí je NOP. Nejméně náročnou instrukcí je pak ADD.

Z obou tabulek je pak zřejmé, že proudový odběr jednotlivých instrukcí závisí také na frekvenci použitého oscilátoru. Jestli se instrukce zpracovává ještě během provádění předcházející instrukce nelze na základě provedených měření přesně určit. Z kapitol 6.7.1 a 6.7.2 plyne, že identifikace instrukce zpracovávané procesorem závisí na:

- tvaru průběhu proudového odběru,
- velikosti jednotlivých proudových špiček,
- datech se kterými konkrétní instrukce pracuje.

Manuální rozpoznání instrukcí je pomalé a ne příliš přesné. To je dáno tím, že odchylky v proudové spotřebě u jednotlivých instrukcí jsou u většiny případů tak malé, že není možné s jistotou určit o kterou instrukci se právě jedná. Hlavní rozdíly mezi jednotlivými instrukcemi jsou pak patrné z rozdílného tvaru průběhu proudového odběru, který je pro každou instrukci charakteristický. Proto by zde mělo význam využít systém pracující na principu umělé inteligence či neuronové sítě. Z výsledných grafů, lze také usuzovat, že celý průběh je ovlivněný nejsložitější instrukcí, která v něm probíhá.

# 7 ZÁVĚR

Cílem diplomové práce bylo realizovat experimentální měření výkonového postranního kanálu mikroprocesoru PIC16F84A, který cyklicky zpracovává jednu libovolnou instrukci.

Nejprve bylo nutné seznámit se s problematikou postranních kanálů, vlastnostmi daného procesoru a jeho instrukcemi, nastudovat principy programování pomocí assembleru, naučit se pracovat ve vývojovém prostředí MPLAB IDE v. 8.33 a vybrat účinnou měřící metodu. Pro měření byla nakonec použita měřící metoda využívající oddělovací transformátor, protože u ní nedocházelo k téměř žádnému nežádoucímu ovlivnění signálu šumem a nedocházelo tak ke zkreslení výsledného průběhu, viz kapitola 6.5.

Poté byly vytvořeny programy XOR, AND, NOP, ADD, INC a SWAP, viz kapitola 6.3. Tyto programy byly postupně nahrány do mikroprocesoru a spuštěny. Následně byl mikroprocesor zapojen do obvodu podle obrázku 5.1. Pomocí osciloskopu se pak sledoval úbytek napětí na rezistoru s odporem 1 Ω při vykonávání jednotlivých programů, které realizovali konkrétní operaci podle dané instrukce. Kvůli eliminaci šumu bylo každé měření opakováno desetkrát a poté zprůměrováno. Měření byla provedena pro oscilátory s taktem 4 a 20 MHz.

Výsledky měření ukázaly, že proudový odběr jednotlivých instrukcí závisí na frekvenci použitého oscilátoru, přičemž pořadí vždy stejných instrukcí seřazených podle proudového odběru v konkrétních situacích může být u každého typu oscilátoru individuální.

Dále bylo zjištěno, že průběh proudové spotřeby je u každé instrukce jiný a přímo ho ovlivňují data, se kterými konkrétní instrukce pracuje, přičemž lze usuzovat, že celkový průběh je ovlivněný nejsložitější instrukcí, která v něm probíhá. Jestli se instrukce začíná zpracovávat ještě během provádění předcházející instrukce se zjistit nepodařilo, ale nelze to s určitostí potvrdit ani vyvrátit, k tomu by byla potřeba další měření.

Rozpoznání instrukce je tedy možné na základě znalosti tvaru průběhu jejího proudového odběru, velikosti jednotlivých proudových špiček a vstupních dat. Ale protože jsou odchylky v proudové spotřebě u jednotlivých instrukcí minimální, nelze identifikaci učinit pouze pomocí lidského oka.

Korektní rozeznání instrukcí by mohlo být uskutečnitelné, kdyby se pro měření použil osciloskop s co možná nejvyšší vzorkovací frekvencí, nebo pomocí systému, který pracuje na principu umělé inteligence či neuronové sítě, což by mohlo být předmětem dalšího výzkumu.

# LITERATURA


[1] *Bagrik* [online]. [2005] [cit. 2009-11-03]. Dostupný z WWW: <www.bagrik.kdhracholusky.cz/Dilny/www/PIC16F84A.doc>.

[2] DANĚČEK, P.: *Útoky na kryptografické moduly*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2007, 122 str. Vedoucí dizertační práce Doc. Ing. Václav Zeman, Ph.D.

[3] *Elektronika E-ZIN* [online]. 2007 [cit. 2009-11-03]. Dostupný z WWW: <http://elektronika.ezin.cz>.

[4] *Historie RISC procesorů* [online]. [2002] [cit. 2009-11-03]. Dostupný z WWW: <http://www.fi.muni.cz/usr/jkucera/pv109/2002/xtoufar1.htm>.

[5] JEDLIČKA, Jiří. *Překlad skript firmy Tektronix*. [s.l.], 2003. 47 s. Oborová práce. Střední průmyslová škola elektrotechnická Praha 2, Ječná 30 . Dostupné z WWW: <http://www.volny.cz/jirka.jedla/XYZ%20Sondy.pdf>.

[6] *Katedra teoretické informatiky a matematické logiky* [online]. [2006] [cit. 2009-11-03]. Dostupný z WWW: <ktiml.mff.cuni.cz/~cepek/PrPo6-Procesory-web.ppt>.

[7] Microchip Technology Inc. *PICDEM™ 2 Plus Demonstration Board User's Guide* [online]. [s.l.] : [s.n.], 2006 [cit. 2010-04-25]. Dostupné z WWW: <http://inst.eecs.berkeley.edu/~ee100/su07/lab/lab8-PROJECT-PIC_Intro/PIC_docs_datasheets/PICDEM_2_Plus_Board_Users_Guide.pdf>.

[8] NOUZÁK, J. *Postranní kanály mikroprocesorů*. Praha, 2007. 48 s. České vysoké učení technické v Praze, Fakulta elektrotechnická. Vedoucí bakalářské práce Ing. Jan Schmidt, Ph.D.

[9] POPOVSKÝ, M. *Útoky postranními kanály*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2009. 71 str. Vedoucí diplomové práce Ing. Zdeněk Martinásek.

[10] PTÁČEK, Luboš. *Power analysis of AES*. Brno, 2008. 34 s. Bakalářská práce. Masarykova univerzita, fakulta informatiky.

[11] *Překlad manuálu k PIC 16F84A* [online]. [2001] [cit. 2009-11-03]. Dostupný z WWW: <http://www.copsu.cz/mikrop/download.php?headr=pic16f84a&plusDB=1>.

[12] *RISC , CISC procesory* [online]. 2006 [cit. 2009-11-03]. Dostupný z WWW: <http://radovan.bloger.cz/IT-internet/RISC---CISC-procesory>.

[13] ŠEMBERA, P. *Analýza DES postranními kanály*. Praha, 2008. 43 s. České vysoké učení technické v Praze, Fakulta elektrotechnická. Vedoucí bakalářské práce Ing. Jan Schmidt, Ph.D.

[14] ŠVENDA, Petr. *Srovnání standardu AES s algoritmy 3DES a IDEA* [online]. [s.l.], 2002. 7 s. Projekt. Masarykova univerzita, fakulta informatiky. Vedoucí práce V. Matyáš. Dostupné z WWW: <http://www.fi.muni.cz/~xsvenda/docs/AEScomparison2001.pdf>.

[15] *Wikipedie* [online]. [2009] [cit. 2009-11-03]. Dostupný z WWW: <http://cs.wikipedia.org>.

# SEZNAM POUŽITÝCH ZKRATEK, VELIČIN A SYMBOLŮ

- **Seznam použitých zkratek.**

| | |
|---|---|
| AES | Pokročilý šifrovací standard (Advanced Encryption Standard) |
| ANOVA | Analýza rozptylu (Analysis Of Variance) |
| DEA, DES | Symetrický blokový šifrovací algoritmus (Data Encryption Standard, Data Encryption Algorithm) |
| CISC | Procesory s komplexní instrukční sadou (Complex Instruction Set Computer) |
| CMOS | Technologie výroby polovodičových součástek (Complementary Metal Oxide Semiconductor) |
| RISC | Procesory s redukovanou instrukční sadou (Reduced Instruction Set Computer) |
| T | Tranzistor |
| USB | Universální sériové rozhraní (Universal Serial Bus) |

- **Seznam použitých veličin.**

| | | |
|---|---|---|
| $C$ | [F] | Elektrická kapacita |
| $I$ | [A] | Elektrický proud |
| $I_c$ | [A] | Nabíjecí (charge) proud |
| $I_d$ | [A] | Vybíjecí (decharge) proud |
| $i(t)$ | [A] | Elektrický proud v čase |
| $f$ | [Hz] | Kmitočet |
| $p(t)$ | [W] | Elektrický příkon v čase |
| $R$ | [Ω] | Elektrický odpor |
| $t$ | [s] | Čas |
| $U$ | [V] | Elektrické napětí |
| $U_{CPU}$ | [V] | Napětí mikroprocesoru |
| $V_{dd}$ | [V] | Kladné napájecí napětí |
| $U_{měřené}$ | [V] | Naměřené napětí |
| $V_{ss}$ | [V] | Napájecí a signálové zem |
| $U_{vst}$ | [V] | Vstupní napětí |
| $U_{výst}$ | [V] | Výstupní napětí |
| $U_{zdroje}$ | [V] | Napětí zdroje |